2

IMACS '91

13TH WORLD CONGRESS ON COMPUTATION AND APPLIED MATHEMATICS

JULY 22 - 26, 1991
TRINITY COLLEGE DUBLIN
IRELAND

PROCEEDINGS

IN FOUR VOLUMES

VOLUME 2

IMACS '91

Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics

July 22-26, 1991,Trinity College, Dublin, Ireland

in four volumes

VOLUME 2

Computational Fluid Dynamics and Wave Propagation
Parallel Computing
Concurrent and Supercomputing
Computational Physics/Computational Chemistry and Evolutionary Systems

EDITED BY: R Vichnevetsky
Rutgers University
New Brunswick, USA

J J H Miller
Trinity College
Dublin, Ireland

IMACS Symposium Rutgers Univ Dept of
Computer Science New Brunswicks, NJ 08903

100.00 per set 4 Vols.

Printed in Ireland by Criterion Press, Dublin

A UNIFORM NUMERICAL METHOD FOR A CLASS OF QUASILINEAR TURNING POINT PROBLEMS

Relja Vulanović
Institute of Mathematics, University of Novi Sad
21000 Novi Sad, Yugoslavia

Abstract. An L^1-stable quasilinear singularly perturbed boundary value problem with a single turning point is solved numerically by a finite-difference scheme on a mesh which is dense near the turning point. The scheme is a special variant of the upwind scheme and it has better properties than the standard Engquist-Osher scheme.

Introduction

We consider the following singularly perturbed boundary value problem:

$$-\varepsilon u'' - xb(x,u)u' + c(x,u) = 0, \quad x \in I = [-1,1], \tag{1}$$

$$u(-1) = U_-, \quad u(1) = U_+, \tag{2}$$

where ε is a small positive parameter, $U_\pm$ are given numbers, b and c are sufficiently smooth functions, and

$$c(x,u) = xc_1(x,u) + \varepsilon c_2(x,u), \tag{3}$$

$$b(x,u) \geq b_* > 0, \quad x \in I, \quad u \in \mathrm{R}. \tag{4}$$

Let

$$f(x,u) = \int_0^u xb(x,s)ds, \quad g(x,v) = f_x(x,u) + c(x,u).$$

Then (1) can be written down in the form.

$$-\varepsilon u'' - f(x,u)' + g(x,u) = 0, \quad x \in I. \tag{5}$$

Furthermore, we assume:

$$g_u(x,u) = c_u(x,u) + (xb(x,u))_x \geq g_* > 0, \quad x \in I, \quad u \in \mathrm{R}. \tag{6}$$

Numerical treatment of problems of this type was considered in [2] (the linear case) and [3] (the semilinear case $b = b(x)$). By using the technique from [2], [3], based on inverse monotonicity and (3), (4), (6), we can get that the problem (1), (2) has a unique solution y and that the following estimates hold for $x \in I$:

$$|y(x)| \leq M, \quad |(xy(x))'| \leq M, \quad |(xy(x))''| \leq M[1 + \mu^{-1}v(x)],$$

$$\varepsilon|y''(x)| \leq M[|x| + \mu + v(x)], \quad \varepsilon|y'''(x)| \leq M[1 + \mu^{-1}v(x)],$$

where $\mu = \varepsilon^{1/2}$ and $v(x) = \exp(-|x|/\mu)$, and throughout M denotes any positive constant independent of ε. These estimates are needed in the consistency-error analysis.

Numerical Method

We shall use finite-differences on a special discretization mesh, the approach from [2], [3]. The estimates above show that y has an interior layer at $x = 0$, and because of that we shall use a mesh which is dense near that point The mesh I^h has the points:

$$x_i = \lambda(hi - 1), \quad i = 0(1)n, \quad h = 2/n, \quad n = 2m, \quad m \in \mathrm{N},$$

$$\lambda(t) = \begin{cases} \omega(t) := \mu t/(1/2 - t) & \text{if} \quad t \in [0, \alpha], \\ \omega'(\alpha)(t - \alpha) + \omega(\alpha) & \text{if} \quad t \in [\alpha, 1], \\ -\lambda(-t) & \text{if} \quad t \in [-1, 0]. \end{cases}$$

Here $(\alpha, \omega(\alpha))$ denotes the contact point of the tangent line from (1,1) to $\omega(t)$.

Let $h_i = x_i - x_{i-1}$, $i = 1(1)n$, and let w^h and z^h denote arbitrary mesh functions defined on $I^h \setminus \{-1,1\}$. We set $w^h = [w_1, w_2, ..., w_{n-1}]^T$ and $w_0 = U_-$, $w_n = U_+$. Furthermore, let

$$D_-w_i = (w_i - w_{i-1})/h_i, \quad D_+w_i = (w_{i+1} - w_i)/h_{i+1},$$

and let $f_i = f(x_i, w_i)$, $g_i = g(x_i, w_i)$. Let us form the discretization of (5), (2):

$$Tw_i := (Tw^h)_i = 0, \quad i = 1(1)n - 1, \tag{7}$$

$$Tw_i = -\varepsilon[D_+w_i - D_-w_i]/h_i - D_-f_i + g_i, \quad i = 1(1)m - 1,$$

$$Tw_i = -\varepsilon[D_+w_i - D_-w_i]/h_{i+1} - D_+f_i + g_i, \quad i = m + 1(1)n - 1,$$

$$Tw_m = -\varepsilon[w_{m-1} - 2w_m + w_{m+1}]/h_m^2 - [f_{m+1} - f_{m-1}]/2h_m + g_m,$$

(note that $h_m = h_{m+1}$). The following stability inequality holds:

$$\|w^h - z^h\|_1 \leq g_*^{-1}\|Tw^h - Tz^h\|_1,$$

with the discrete L^1 norm:

$$\|w^h\|_1 = \sum_{i=1}^{m} h_i|w_i| + \sum_{i=m+1}^{n-1} h_{i+1}|w_i|.$$

Then the first order uniform convergence can be proved due to the special mesh and the estimates from the Introduction:

$$\|y^h - y_h\|_1 \leq Mh,$$

where M does not depend on h, y^h is the unique solution to (7) and y_h is the restriction of y on $I^h \setminus \{-1,1\}$. Moreover, numerical results show pointwise uniform convergence as well. This is not the case with the Engquist-Osher (EO) scheme [1] in general, see [2]. The EO scheme is uniformly convergent only globally – in the standard discrete L^1 norm [1-3]. This is because the EO scheme uses $D_\pm$ with $(h_i + h_{i+1})/2$ instead of h_{i+1} and h_i. Another advantage of our scheme is that it uses a simpler mesh generating function λ than the upwind schemes in [2] and [3].

We illustrate these facts by some numerical results for the problem with $b = 1$, $c = -\pi[x\sin(\pi x) + \varepsilon\pi\cos(\pi x)]$ and $U_- = -2$, $U_+ = 0$, for which the solution is known, see [3]. Let E denote the maximal pointwise error for $\varepsilon = 10^{-12}$. Our scheme gives $E = 0.386, 0.215, 0.115$ for $n = 50, 100, 200$, respectively. Results for other values of ε are similar due to the special mesh. The EO scheme does not converge: $E = 4.99, 8.131, 8.09$ for the same values of n.

References

[1] L. Abrahamsson and S. Osher, Monotone difference schemes for singular perturbation problems, *SIAM J. Numer. Anal.* 19 (1982), 979-992.

[2] R. Vulanović, On numerical solution of a turning point problem, *Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mat.* 19, 1 (1989), 11-24.

[3] R. Vulanović, On numerical solution of a mildly nonlinear turning point problem, *RAIRO Math. Model. Numer. Anal.* 24 (1990), 765-784.

A NOTE ON A SPLINE COLLOCATION METHOD FOR SINGULARLY PERTURBED PROBLEMS

Katarina Surla
Institute of Mathematics, University of Novi Sad
21000 Novi Sad, Yugoslavia

Zorica Uzelac
Faculty of Technical Sciences, University of Novi Sad
21000 Novi Sad, Yugoslavia

Abstract: The exponential spline collocation methods for singularly perturbed boundary value problem are considered. The convergence between mesh points for different collocation conditions is compared. Numerical results are presented.

Introduction

We consider the following singularly perturbed boundary value problem:

$$\varepsilon y'' + p(x)y' = f(x), \quad x \in I = [0,1], \qquad (1)$$

$$y(0) = \alpha_0, \quad y(1) = \alpha_1, \qquad (2)$$

where ε is a small positive parameter, α_0 and α_1 are given numbers, p and f are sufficiently smooth functions and $p(x) \geq p > 0$. By using exponential spline $e(x)$ from [2], $e(x) \in C^1(I)$, as a collocation function a family of difference schemes is derived in [4]. The well known Allan-Southwel- Il'in and El Mistikawy-Werle (EMW) schemes are members of this family. Some of the properties of the scheme (4),(5),(6) , which belongs to the same family, are better than those of EMW scheme ([3]). In this paper we consider the approximation between the mesh points which correspond to both EMW scheme and to scheme (4),(5),(6). Both splines have first order of uniform convergence on the whole interval, but the numerical results are much better for the scheme (4),(5),(6). This is the consequence of the better accuracy at the nodes. It will also be shown that the collocation spline given in [4] is equal to the one used for the derivation of EMW scheme ([1]).

Collocation Method

The spline e(x) has the form ([2]):

$$e(x) = e_j(x) = u_j + hm_j t + g_j(ch\mu_j t - 1)/\rho_j + q_j(sh\mu_j t - \mu_j)/\rho_j,$$

$$x \in [x_j, x_{j+1}],$$

where t=(x-x$_j$)/h, x=jh, h=1/(n+1), $\mu_j = h\rho_j$, j=0(1)n, ρ_j are tension parameters. The values g_j and q_j are determined from the requirement $e(x) \in C^1(I)$ From the collocation conditions

$$\varepsilon e''(x) + p^+ e'(x) = f^+, \quad x = x_j, x = x_{j+1}, \qquad (3)$$

where p^+ and f^+ are constant aproximations to $p(x)$ and $f(x)$ on the interval $[x_j, x_{j+1}]$ for fixed j, the following family of the difference schemes is derived in [4]:

$$r^- u_{j-1} + r^c u_j + r^+ u_{j+1} = q^- f^- + q^+ f^+, \quad j = 1(1)n, \qquad (4)$$

$$u_0 = \alpha_0, \quad u_1 = \alpha_1, \qquad (5)$$

where

$$r^+ = (\rho^+/(1 - exp(-\mu^+), \quad r^- = \rho^- exp(-\mu^-)/(1 - exp(-\mu^-)),$$

$$r^c = -r^- - r^+$$

$$q^+ = (exp(-\mu^+) + \mu^+ - 1)/(p^+(1 - exp(-\mu^+))),$$

$$q^- = (1 - exp(-\mu^-) - \mu^- exp(-\mu^-)/(p^-(1 - exp(-\mu^-))),$$

$\mu^+ = \rho^+ h, \mu^- = \rho^- h, \rho^+ = p^+/\varepsilon, \rho^- = p^-/\varepsilon, p^-$ and f^- are constant aproximations to $p(x)$ and $f(x)$ on the interval $[x_{j-1}, x_j]$. By determing $p^\pm = (p(x_{j\pm1}) + p(xj))/2, f^\pm = (f(x_{j\pm1}) + f(x_j))/2$ we obtain EMW scheme, whereas for

$$p^\pm = p(x_j \pm h/2), \quad f^\pm = f(x_j \pm h/2) \qquad (6)$$

we obtain the scheme from [3] ((4),(5),(6)). In [1] the exact solution of the problem:

$$\varepsilon u'' + p^+ u' = f^+, \quad x_j < x < x_{j+1},$$

$$u(x_j) = u_j, \quad u(x_j) = u_{j+1},$$

is used for the approximation between mesh points. The function $e_j(x)$ has the form:

$$e_j(x) = span\{1, x, exp(\rho_j x), exp(-\rho_j x)\}.$$

Since conditions (3) lead to the elimination of the function $exp(\rho_j x)$ from the spline base, after some analysis of the constants one can see that $u(.) = e_j(x)$. Thus, the calculation becomes simpler when one use the spline in the form of the piecewise function $u(x)$. The properties of that function are given in [2]. Some connections between the spline e(x) and the cubic spline are given in [6]. The family of difference schemes corresponding to the cubic spline is presented in [5]. With regard to second order polinomials some characteristics of scheme (4),(5),(6) are shown in [3]: the scheme becomes exat when ε goes to zero ; the major term of the error is four times smaller than the one for EMW scheme when $h \leq \varepsilon$. Both schemes have the second order of uniform convergence at the mesh points. The presented numerical results show that the error between mesh points has similar properties. In the way presented in [1] one can prove that the estimates given for spline corresponding to EMW scheme are also valid for the spline corresponding to scheme (4), (5),(6).

Numerical results

The example is taken from [1]. We denote by E_n the maximum of $|y(x_j + h/2) - u(x_j + h/2)|$, $j = 0(1)n$. The order of convergence (Ord) for two succesive values of n with respective errors E_n and E_{2n} is defined in the usual way as in [1]. Tables 1 and 2 present the numerical results obtained by EMW and scheme (4),(5),(6) respectively. The better behaviour for small ε of the scheme presented in Table 2 results in hasty decline of Ord.

n	ε				
	2^{-1}	2^{-5}	2^{-10}	2^{-14}	
8	5.578(-3)	8.314(-3)	3.373(-3)	3.153(-3)	E_n
					Ord
16	1.393(-3)	1.891(-3)	9.356(-4)	8.176(-4)	E_n
					Ord
32	3.491(-4)	4.865(-4)	2.940(-4)	2.095(-4)	E_n
	1.985	1.856	1.924	1.930	*Ord*
64	8.733(-5)	1.221(-4)	3.055(-4)	5.378(-5)	E_n
	2.000	1.976	1.938	1.966	*Ord*
128	2.184(-5)	3.070(-5)	2.173(-4)	1.400(-5)	E_n
	2.000	1.956	1.876	1.983	*Ord*
256	5.459(-6)	7.687(-6)	4.800 (-5)	3.774(-6)	E_n
	2.000	1.993	1.687	1.990	*Ord*
512	1.365(-6)	1.922(-6)	5.870(-6)	2.549(-6)	E_n
	2.000	1.998	1.633	1.989	*Ord*

Table 1

n	ε				
	2^{-1}	2^{-5}	2^{-10}	2^{-14}	
8	1.364(-3)	2.049(-3)	1.045(-3)	1.070(-3)	E_n
					Ord
16	3.464(-4)	4.040(-4)	2.585(-4)	2.685(-4)	E_n
					Ord
32	8.716(-5)	1.152(-4)	6.360(-5)	6.663(-5)	E_n
	1.967	1.138	2.011	1.945	*Ord*
64	2.183(-5)	3.006(-5)	2.085(-4)	1.650(-5)	E_n
	1.996	1.758	2.109	1.980	*Ord*
128	5.459(-6)	7.642(-6)	1.891(-4)	4.084(-6)	E_n
	1.999	1.873	.836	2.004	*Ord*
256	1.365(-6)	1.920(-6)	4.172 (-5)	1.010(-6)	E_n
	2.000	1.965	-1.186	2.031	*Ord*
512	3.412(-7)	4.805(-7)	4.771(-6)	1.268(-6)	E_n
	2.000	1.998	0.252 (-1)	2.075	*Ord*

Table 2

References

[1] A. Berger, J. Solomon and M. Ciment, An Analysis of a Uniformly Accurate Difference Method for a Singular Perturbation Problem, *Math. Comput.* 37 (1981) 79-94.

[2] W. Hess and J.W.Schmidt, Convexity Preserving Interpolation with Exponential Splines, *Computing* 36 (1986) 335-342.

[3] K. Surla and Z. Uzelac, An Analysis and Improvement of El Mistikawy and Werle scheme (to appear).

[4] K. Surla and Z. Uzelac, A Family of Spline Difference Schemes, *ZAMM* (to appear).

[5] K Surla and Z. Uzelac, Some uniformly convergent spline difference schemes for singularly perturbed boundary value problem, *IMA J. Numer. Anal.* 10 (1990) 209-222.

[6] K. Surla and Z. Uzelac, The spline collocation method for boundary value problem, Proceedings of the Conference ISAM'91 (to appear).

A Posteriori Error Bounds for Piecewise Linear Approximate Solutions of Singularly Perturbed Nonlinear Elliptic Problems

Koichi Niijima
Department of Control Engineering and Science
Kyushu Institute of Technology
Iizuka 820, Japan

Abstract: A method for finding a posteriori error bounds for piecewise linear approximate solutions of singularly perturbed nonlinear elliptic problems is proposed. A relation between a line integral on an edge of a triangle and volume integrals in the triangle plays an important role.

1. Introduction

Recently, we developed a method for finding error estimators for piecewise linear approximate solutions of nonlinear elliptic problems (Niijima [1], Niijima [2]). We will apply this method to piecewise linear approximate solutions of singularly perturbed nonlinear elliptic problems. Generally, numerical solutions of such problems do not necessarily have a continuous piecewise linear form. So numerical data obtained are interpolated piecewise linearly such that our method can be applied.

2. Preliminaries

Let Ω be a bounded polyhedral domain in R^2 with a boundary $\partial\Omega$ and consider the following problem:

$$-\varepsilon\Delta u + f(x,y,u,\nabla u) = 0 \quad in\,\Omega, \qquad (1)$$
$$u = 0 \quad on\,\partial\Omega, \qquad (2)$$

where ε is a sufficiently small positive constant and $\nabla u = (u_x, u_y)$.

A weak form of (1) and (2) is

$$\varepsilon(\nabla u, \nabla v) + (f(x,y,u,\nabla u), v) = 0 \qquad (3)$$

for any v belonging to the Hilbert space $H_0^1(\Omega)$, where $(\cdot,\cdot)$ denotes an $L^2(\Omega)$ inner product.

We assume that

H1. (3) has a solution in $H_0^1(\Omega)$,

H2. there exists $\alpha > 0$ such that for $v, w \in H_0^1(\Omega)$,

$$\varepsilon(\nabla(v-w), \nabla(v-w)) + (f(x,y,v,\nabla v) - f(x,y,w,\nabla w), v-w) \geq \varepsilon \,\|\, \nabla(v-w) \,\|^2 + \alpha \,\|\, v-w \,\|^2,$$

where $\|\cdot\|$ indicates an $L^2(\Omega)$ norm.

Consider a triangulation of Ω and let F be the set of triangles. Let τ be a triangle in F and let three edges of τ be γ_1, γ_2 and γ_3. Denote the vertices corresponding to γ_1, γ_2 and γ_3 by (x_1,y_1), (x_2,y_2) and (x_3,y_3), respectively.

We have the following lemma.

Lemma 1. For g belonging to the Hilbert space $H^1(\tau)$, we have

$$\langle 1, g\rangle_{\gamma_3} = \frac{|\gamma_3|}{|J_\tau|}[2(1,g)_\tau + (x-x_3, g_x)_\tau + (y-y_3, g_y)_\tau].$$

Here $<\cdot,\cdot>_{\gamma_3}$ and $(\cdot,\cdot)_\tau$ indicate inner products on γ_3 and τ, respectively. Also

$$J_\tau = (x_2-x_1)(y_3-y_1) - (x_3-x_1)(y_2-y_1)$$

and $|\gamma_3|$ denotes the length of γ_3.

This relation is a formula changing a line integral on an edge into the sum of three volume integrals in τ. By this formula, we can rewrite the line integrals appearing in partial integrations of the gradient term by elementwise volume integrals.

3. Main results

Let E be the set of edges not on $\partial\Omega$. Consider two triangles τ_- and τ_+ sharing an edge γ in E, where a normal direction n is outward from τ_-. Let (x_1,y_1), (x_2,y_2) and (x_-,y_-) be the vertices of τ_-, and (x_1,y_1), (x_2,y_2) and (x_+,y_+) the vertices of τ_+. Denote the mesh size by h. Let u^h be a continuous piecewise linear function and define a jump in $\partial u^h/\partial n$ across γ by

$$[\frac{\partial u^h}{\partial n}]_\gamma = \frac{\partial u^h}{\partial n}|_{\tau_+} - \frac{\partial u^h}{\partial n}|_{\tau_-}.$$

For latter convenience. we define $[\partial u^h/\partial n]_\gamma = 1$ for edges γ on $\partial\Omega$.

Let τ be a triangle in F and let γ_1, γ_2 and γ_3 be three edges of τ. From now, we use the symbol u^h to denote a piecewise linear interpolate solution of (3). We now define an operator Δ_τ^h by

$$\Delta_\tau^h u^h = \frac{2}{|J_\tau|}\sum_{i=1}^{3} w_{\tau,i}|\gamma_i|[\frac{\partial u^h}{\partial n}]_{\gamma_i}.$$

Here, if $\gamma_i \in E$, then the parameter $w_{\tau,i}$ has a relation

$$w_{\tau,i} + w_{\tau',i'} = 1$$

for the parameter $w_{\tau',i'}$, where τ' is the other triangle sharing γ_i. If γ_i is on $\partial\Omega$, then $w_{\tau,i}$ is free and $[\partial u^h/\partial n]_{\gamma_i} = 1$.

Using the same symbols as above, we further define a two-dimensional vector r_τ^h by

$$r_\tau^h = \frac{1}{|J_\tau|}(\sum_{i=1}^{3} w_{\tau,i}|\gamma_i|[\frac{\partial u^h}{\partial n}]_{\gamma_i}(x - x_i),$$
$$\sum_{i=1}^{3} w_{\tau,i}|\gamma_i|[\frac{\partial u^h}{\partial n}]_{\gamma_i}(y - y_i)).$$

We define Δ^h and r^h by $\Delta^h = (\Delta_\tau^h)_{\tau\in F}$ and $r^h = (r_\tau^h)_{\tau\in F}$, respectively. By W, we denote the set of vectors whose components consist of all w_τ.

Using Lemma 1, we can prove

Lemma 2. Let u be a solution of (3) and let u^h be a piecewise linear interpolate solution of (3). We put $e = u - u^h$ and define L by

$$L = \varepsilon(\nabla e, \nabla e)$$
$$+(f(x,y,u,\nabla u) - f(x,y,u^h,\nabla u^h), e).$$

Then we have

$$L = -(-\varepsilon\Delta^h u^h + f^h, e) + (\varepsilon r^h, \nabla e),$$

where we put $f^h = f(x,y,u^h,\nabla u^h)$ for simplicity.

We obtain the following theorem by applying the Schwarz' inequality to the right hand side of L in Lemma 2.

Theorem. We have, for $e = u - u^h$,

$$\varepsilon\|\nabla e\|^2 + \alpha\|e\|^2$$
$$\leq \inf_W\{\frac{1}{\alpha}\| -\varepsilon\Delta^h u^h + f^h\|^2 + \varepsilon\|r^h\|^2\}. \quad (4)$$

Remark: $\inf_W\{\frac{1}{\alpha}\| -\varepsilon\Delta^h u^h + f^h\|^2 + \varepsilon\|r^h\|^2\}$ is a quadratic minimization problem.

4. Numerical results

Example.

$$-\varepsilon\Delta u + u^3 + u - g = 0$$
$$in\ \Omega = (0,1)\times(0,1),$$
$$u = 0\ \ on\ \partial\Omega,$$

where $\varepsilon = 10^{-3}$ and g is determined such that $u = (1 - exp(-\frac{x(1-x)}{\sqrt{\varepsilon}}))(1 - exp(-\frac{y(1-y)}{2\sqrt{\varepsilon}}))$ satisfies the above equation. It is easily verified that H2 holds as $\alpha = 1$. Divide the interval $(0,1)$ into m-equidistant subintervals and make a triangulation. The numerical solutions u^h were obtained by the Ritz-Galerkin method. A posteriori error bounds were computed following Theorem, and were compared with actual errors.

m	$\sqrt{r.h.s\ of\ (4)}$	$\sqrt{l.h.s\ of\ (4)}$
4	0.732	0.402
6	0.486	0.281
8	0.353	0.214
10	0.274	0.170
12	0.223	0.142
14	0.188	0.121
16	0.163	0.105

The experiment was performed by using Turbo Pascal Ver.5.5 on the personal computer EPSON PC-286UX.

Refereces

[1] K.Niijima, A posteriori error bounds for piecewise linear approximate solutions of nonlinear elliptic equations, to appear in Math. of Comp.

[2] K.Niijima, A posteriori error bounds for piecewise linear approximate solutions of regularized compressible flow problems, to appear in Numer. Math.

Iterative methods for convection dominated flow*

R. B. Kellogg
Inst. Physical Sci. Tech.
University of Maryland
College Park, Md. 20742 USA

Abstract - Some iterative methods are considered for the numerical solution of convection diffusion problems. The first class of iterative methods is Chebyshev accelerated iterations. The issues of parameter selection and convergence rates are considered. Secondly, we consider convection - diffusion type iterations where the iterations are of a Peaceman-Rachford type. Here, a convergence method is established, and a conjecture is given concerning a related problem in functional analysis.

A. Chebyshev iterations

We consider Chebyshev accelerated iterations for the numerical solution of discretizations of the convection diffusion equation

$$\begin{aligned} -\epsilon\Delta u + pu_x + qu_y + ru &= f \text{ in } \Omega, \\ u &= y \text{ on } \partial\Omega, \end{aligned} \tag{1}$$

and related systems, such as the Oseen system. If a discretized version of (1) is written

$$Au = f, \tag{2}$$

the methods we consider may be written

$$u^{k+1} = \alpha_k A u^k + \beta_k u^k + \gamma_k u^{k-1}, \tag{3}$$

with initial guess u^0, where the iteration parameters $\alpha_k, \beta_k, \gamma_k$ satisfy

$$\beta_k + \gamma_k = 1, \quad \gamma_0 = 0. \tag{4}$$

From (3) and (4) one finds that the solution u is preserved under the iteration, and the error $e^k = u - u^k$ satisfies $e^{k+1} = \alpha_k A e^k + \beta_k e^k + \gamma_k e^{k-1}$. Hence, defining a set of polynomials $P_k(\lambda)$ by

$$P_0(\lambda) = 1, \; P_{k+1}(\lambda) = \alpha_k \lambda P_k(\lambda) + \beta_k P_k(\lambda) + \gamma_k P_{k-1}(\lambda), \; k = 0, 1,$$

we find that $e^k = P_k(A)e^0$. From this formula it is seen that the iteration parameters should be chosen so that the values of P_k are small on the spectrum of A. Manteuffel [1] has shown how to choose the P_k in terms of Chebyshev polynomials so as to optimize the convergence of the iterations. Manteuffel's choice requires a knowledge of an ellipse $\mathcal{E}$ that contains the spectrum of A and that does not contain the origin.

In the first part of this talk we show how to obtain ellipses $\mathcal{E}$ in an explicit manner from a knowledge of the coefficients of the equation (1), the mesh spacing h, and the discretization A. We also give estimates for the asymptotic rate of convergence of the resulting iterative method in terms of the parameters ϵ and h. Finally, we give similar results for a preconditioned version of (1), where we precondition by the self adjoint part of the operator.

Details of this work are contained in [2].

B. Peaceman Rachford type iterations

We consider the convection diffusion equation

$$\begin{aligned} Lu \equiv -\Delta u + \mathbf{p}\cdot\nabla u + ru &= f, \quad \text{in } \Omega, \\ u = 0 \text{ on } \Gamma &= \partial\Omega. \end{aligned} \tag{4}$$

Divide Γ into two subsets, $\Gamma_{in} = \{(x,y) . \mathbf{n} \;\; \mathbf{p} < 0\}$, and $\Gamma_{out} = \Gamma \setminus \Gamma_{in}$. Here $\mathbf{n}$ is the outward point unit normal to Γ. We consider the CDI method for solving (i): guess u^0, and define $u^{1/2}, u^1, \;\;$, by

$$-\Delta u^{k+1/2} + \rho u^{k+1/2} = \rho u^k - \mathbf{p}\cdot\nabla u^k - ru^k + f, \quad u^{k+1/2} = 0 \text{ on } \Gamma,$$
$$\mathbf{p}\cdot\nabla u^{k+1} + \rho u^{k+1} = \rho u^{k+1/2} - Du^{k+1/2} + f, \quad u^{k+1} = 0 \text{ on } \Gamma_{in}.$$

To study the convergence of this method, it is convenient to define the operator L_D by

$$L_D u \equiv -\Delta u, \quad u = 0 \text{ on } \Gamma,$$

and the operator L_C by

$$L_C u \equiv \mathbf{p}\cdot\nabla u + ru, \quad u = 0 \text{ on } \Gamma_{in}.$$

With these definitions, the iterations may be written

$$\begin{aligned} (\rho I + L_D)u^{k+1/2} &= (\rho I - L_C)u^k + f, \\ (\rho I + L_C)u^{k+1} &= (\rho I - L_D)u^{k+1/2} + f. \end{aligned} \tag{6}$$

We regard L_D and L_C as closed, unbounded operators on $L_2(\Omega)$. L_D and L_C are accretive in the sense that for some $\alpha > 0$,

$$\begin{aligned} (L_D u, u) &> \alpha(u,u), \\ (L_C u, u) &> \alpha(u,u). \end{aligned} \tag{7}$$

Also, L_D^{-1} and L_C^{-1} are bounded operators on $L_2(\Omega)$, and L_D^{-1} is a compact operator. Finally, any positive ρ is in the resolvant set of L_D and L_C. With this understanding we define

$$\begin{aligned} v^{k+1/2} &= (\rho I + L_D)u^{k+1/2}, \quad k = 0, 1, \cdots, \\ v^k &= (\rho I + L_C)u^k, \quad k = 0, 1, \cdots, \end{aligned} \tag{8}$$

and we set

$$\begin{aligned} T_D &= (\rho I + L_D)^{-1}(\rho I - L_D) = 2\rho(\rho I + L_D)^{-1} - I, \\ T_C &= (\rho I + L_C)^{-1}(\rho I - L_C) = 2\rho(\rho I + L_C)^{-1} - I. \end{aligned} \quad (9$$

Thus, T_D and T_C are bounded operators, and T_D is a compact perturbation of $-I$. Also, from the accretiveness of L_C anmd L_D one can show that

$$\|T_D\| \le 1, \quad \|T_C\| \le 1.$$

In terms of these operators, (2) may be written

$$v^{k+1/2} = T_C v^k + f,$$
$$v^{k+1} = T_D v^{k+1/2} + f.$$

To establish the convergence of the method, we must show that

$$(T_D T_C)^k w \to 0$$

for any $w \in L_2(\Omega)$. This is easily shown in the finite dimensional case. It becomes an interesting conjecture in the case of the differential operator.

References

1. T. A. Manteuffel, "The Tchebychev iteration for nonsymmetric matrices", Numer. Math. 28(1977), 307-327.

2. R. B. Kellogg, "Spectral bounds and iterative methods in convection dominated flow", to appear.

*Supported in part by the U. S. Army Research Office

EXPLICIT FINITE ELEMENT METHODS FOR CONVECTION-DIFFUSION PROBLEMS

GERARD R. RICHTER
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract. We describe some recent work on explicit finite element methods for convection dominated convection-diffusion problems. We develop the methods for pure hyperbolic equations, and then discuss their extension to problems with diffusion.

1. Introduction. Our purpose is to summarize some recent work on finite element methods for convection-diffusion equations in which convection is the dominant term. One alternative for such problems is the streamline diffusion method [3,4], in which the usual Galerkin's method test functions are augmented by a convective derivative term. There are also "explicit" finite element methods, which permit development of an approximation in an element by element, as opposed to global, fashion. It is the latter class of methods that we shall be concerned with.

We first describe these methods for a linear, scalar hyperbolic problem

$$\text{(1.1)}\qquad \begin{aligned} &\alpha(x)\cdot\nabla u+\beta(x)u=f(x), &&x\in\Omega,\\ &u=g, \qquad x\in\Gamma_{in}(\Omega). \end{aligned}$$

Here $\Omega\subset R^2$ is a bounded polygon with boundary Γ, and α is assumed to have unit length. The "inflow" boundary $\Gamma_{in}(\Omega)\subset\Gamma$ is characterized by $\alpha\cdot n<0$ where n is the unit outer normal to Ω.

We shall assume Ω has been divided into triangles and/or rectangles in such a way that the nonalignment condition $|\alpha\cdot n|\neq 0$ holds for all element edges. This amounts to an assumption of unidirectional "flow" across all edges, and allows the elements to be ordered explicitly with respect to domain of dependence [6]. In other words, the solution of the continuous problem (1.1) can be developed first in one element (the inflow to which must be contained in $\Gamma_{in}(\Omega)$), then in another, etc. In general there will be many explicit orderings for a given mesh, and it is potentially advantageous to view the solution as evolving as a front, in parallel, across layers of elements. The class of finite element methods of interest here are those which allow development of an approximate solution in the same explicit manner. Henceforth we shall deal exclusively with the case of triangular elements.

We need some additional notation to describe these methods. For a generic triangle T, let $\mathrm{P}_n(T)$ denote the set of polynomials of degree $\le n$ over T, i. e., linear combinations of $x^iy^j, 0\le i+j\le n$. We denote by S_h^0 the space of piecewise polynomials over the given triangulation whose restrictions to individual triangles T lie in $\mathrm{P}_n(T)$. A function $w_h\in S_h^0$ will in general be discontinuous across triangle edges Γ_i, and for $P\in\Gamma_i$ we define its upstream (–) and downstream (+) limits by $w_h^\pm(P)=\lim_{\epsilon\to0^+}w_h(P\pm\epsilon\alpha)$. The space $S_h^1\subset S_h^0$ will consist of continuous piecewise polynomials over the same triangulation.

The discontinuous Galerkin method [6,7] produces an approximation $u_h\in S_h^0$ satisfying the conditions

$$\text{(1.2)}\qquad \begin{aligned} (\alpha\cdot\nabla u_h+\beta u_h,v_h)-\int_{\Gamma_{in}(T)}(u_h^+-u_h^-)v_h\,\alpha\cdot n\,d\tau\\ =(f,v_h),\qquad \text{all } v_h\in\mathrm{P}_n(T). \end{aligned}$$

Here (,) is the $L^2(T)$ inner product and τ denotes arclength along the boundary of T. The approximate solution u_h starts off as an interpolant of the given inflow data g, and is propagated, triangle by triangle, via the above inner product conditions. The triangles must of course be processed in an explicit order.

To formulate a continuous analog of the discontinuous Galerkin method, we need to distinguish between one-inflow-side (type I) triangles and two-inflow-side (type II) triangles. In developing $u_h\in S_h^1$ in a type I (type II) triangle T, note that u_h will have $n+1$ $(2n+1)$ fewer degrees of freedom in T because u_h, now continuous, is known already on $\Gamma_{in}(T)$. Thus it is natural to define a continuous approximation $u_h\in S_h^1$ via the conditions [7]:

$$\text{(1.3)}\qquad (\alpha\cdot\nabla u_h+\beta u_h,v_h)=(f,v_h),\qquad \text{all } v_h\in V_h,$$

where $V_h=\mathrm{P}_{n-1}(T)$ if T is of type I and $V_h=\mathrm{P}_{n-2}(T)$ if T is of type II. This will give equality between the number of equations and unknowns in each triangle. We assume $n\ge 2$ for method (1.3), so that the inner product conditions are nonvacuous for both types of triangles. As before, u_h starts as an interpolant of g on $\Gamma_{in}(\Omega)$.

Both of these finite element methods are generalizations of the most basic first order upwind finite difference scheme.

They share its good stability properties, and may be applied for arbitrarily large n. Numerical computations typically show the optimal $O(h^{n+1})$ rate of convergence when the solution u is sufficiently smooth. Theoretical error estimates may be found in [1,5,6,8]. Moreover, the methods yield good results when applied to problems with discontinuous solutions [11]. It can be shown [2] that the influence of a disturbance propagating along a characteristic is confined to a band of width approximately $O(\sqrt{h})$ about the characteristic. Thus as $h \to 0$, the methods methods exhibit the right limiting domain of dependence behavior.

2. Convection-diffusion equations.

We now consider a convection-diffusion equation

$$
\begin{aligned}
(2.1) \qquad & -\epsilon\Delta u + \alpha\cdot\nabla u + \beta u = f, \qquad x \in \Omega, \\
& u = g, \qquad x \in \Gamma,
\end{aligned}
$$

where α, as before, has unit length, and ϵ is a positive constant. If ϵ is large in comparison to the mesh size h, then diffusion will be the dominant transport term, and the standard Galerkin method performs well. However, if $\frac{\epsilon}{h}$ is small, convection will be dominant, and Galerkin's method is no longer the finite element method of choice. Solution features that are not resolved generate oscillations which tend to propagate throughout the domain. The methods (1.2) and (1.3) can be extended to convection dominated problems of the form (2.1), thus providing an alternative.

The discontinuous Galerkin method can be extended to (2.1) by treating the diffusion term in essentially the same way as the convection term:

(2.2)

$$
(-\epsilon\Delta u_h + \alpha\cdot\nabla u_h + \beta u_h, v_h) + \int_{\Gamma_{in}(T)} \epsilon\left(\frac{\partial u_h^+}{\partial n} - \frac{\partial u_h^-}{\partial n}\right) v_h\, d\tau
$$

$$
- \int_{\Gamma_{in}(T)} (u_h^+ - u_h^-) v_h\, \alpha\cdot n\, d\tau = (f, v_h), \qquad \text{all } v_h \in P_n(T).
$$

This scheme uses upstream values of both u_h and its normal derivative. (A minor modification needs to be mode if $\Gamma_{in}(T) \cap \Gamma_{in}(\Omega) \neq \emptyset$ because $\frac{\partial u_h^-}{\partial n}$ is available only in the interior of Ω.) The approximation u_h starts as an interpolant of g on $\Gamma_{in}(\Omega)$, and is developed in the same explicit manner as in the pure hyperbolic case. It is shown in [12] that this scheme is stable for sufficiently small $\frac{\epsilon}{h}$, assuming the triangle sides (with the possible exception of those on Γ) are uniformly bounded away from the characteristic direction. Near optimal error estimates are also derived there. Similar extensions of the continuous method (1.3) are developed in [9,10].

For a convection-diffusion problem in which convection is dominant only over part of Ω, these hyperbolic-based finite element methods can be applied locally, in conjunction with the standard Galerkin method. For example, the solution to (2.1) typically has an outflow boundary layer of thickness $O(\epsilon)$ [13]. One could apply (2.2), say, up to this point and then use Galerkin's method over a finer mesh to resolve the outflow boundary layer.

Extensions of the methods (1.2) and (1.3) to equations with a nonlinear convection term are currently under study.

REFERENCES

[1] R. S. Falk and G. R. Richter, *Analysis of a continuous finite element method for hyperbolic equations*, SIAM J. Numer. Anal., 24 (1987), pp. 257–278.

[2] R. S. Falk and G. R. Richter, *Local error estimates for a finite element method for hyperbolic and convection-diffusion equations*, *preprint.*

[3] T. J. R. Hughes and A. Brooks, *A multidimensional upwind scheme with no crosswind diffusion*, in Finite Element Methods for Convection Dominated Flows, T. J. R. Hughes, ed., ASME, New York, 1979.

[4] C. Johnson, U. Nävert, and J. Pitkäranta, *Finite element methods for linear hyperbolic problems*, Computer Methods in Applied Mechanics and Engineering, 45 (1984), pp. 285–312.

[5] C. Johnson and J. Pitkäranta, *An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation*, Math. Comp., 46 (1986), pp. 1–26.

[6] P. Lesaint and P. A. Raviart, *On a finite element method for solving the neutron transport equation*, Mathematical Aspects of Finite Elements in Partial Differential Equations, C. deBoor, ed., Academic Press, New York, 1974, pp. 89–123.

[7] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation*, Los Alamos Scientific Laboratory Report LA-UR-73-479 (1973), Los Alamos, New Mexico.

[8] G. R. Richter, *An optimal-order error estimate for the discontinuous Galerkin method*, Math. Comp., 50 (1988), pp. 75–88.

[9] G. R. Richter, *A finite element method for time dependent convection-diffusion equations*, Math. Comp., 54 (1990), pp. 81–106.

[10] G. R. Richter, *An explicit finite element methos for convection dominated steady state convection-diffusion equations*, SIAM J. Numer. Anal., to appear.

[11] G. R. Richter, *Explicit finite element for scalar hyperbolic equations*, Appl. Num. Math, to appear.

[12] G. R. Richter, *The discontinuous Galerkin method with diffusion*, *preprint.*

[13] M. I. Vishik and L. A. Lyusternik, *Regular degeneration and boundary layer for linear differential equations with a small parameter*, Uspekki Mat. Nauk., 12 (1957), pp. 3–122; Amer. Math. Soc. Transl., Ser. 2, 20 (1962), pp. 239–364.

ON THE DETERMINATION OF THE ORDER OF UNIFORM CONVERGENCE

Paul A. Farrell
Department of Mathematics & Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Alan Hegarty
Department of Mathematics
University of Limerick
Limerick, Ireland.

Abstract:- We shall discuss a number of methods used in the literature to calculate rates of uniform convergence. We mention some anomalies concerning interpretation of the resulting tables and rates, which lead to the determination of experimental rates of uniform convergence lower than the correct rates.

Introduction

The determination of the order of uniform convergence is not always a straightforward task. A number of approaches exist in the literature, the two major variants being that appearing in [6] and [1] and that in [2].

The former approach involves solving numerically a singularly perturbed differential equation on $[x_L, x_R]$ for which the analytic solution is known. The difference equation is solved for decreasing values of h and the rate of convergence calculated from

$$p_{\epsilon\epsilon}^h = (\ln e_{\epsilon\epsilon}^{2h} - \ln e_{\epsilon\epsilon}^h)/\ln(2) \tag{1}$$

where

$$e_{\epsilon\epsilon}^h = \max_{0\leq i\leq N} |u_i^h - u_\epsilon(x_i)|,\ h = [x_L - x_R]/N. \tag{2}$$

The equation solved is chosen so that the solution and its derivatives exhibit exactly the analytic behaviour hypothesized in the proof of the error estimates. In practise this is achieved by choosing a solution and then determining a differential equation of the correct form which this satisfies.

The uniform rate is determined by inspecting a table of values of $p_{\epsilon\epsilon}^h$, for varying h and ϵ, constructed by setting $\epsilon = h^s$ for various values of s. Results of this form are given in [6] for a non-turning point problem and in [1] for a turning-point problem of the type considered in [3, 4]. In the case of the turning-point problem the choice of solution involves making the boundary values functions of ϵ and hence by the choice of $\epsilon = h^s$ functions of h. A disadvantage of this method is the requirement of prior knowledge of the solution since this limits the ease with which it can be applied to other problems. An alternative is, of course, to use an accurate approximation on a *fine* mesh, if this is determinable.

The Double Mesh Method

The method proposed in [2] is based on a consequence of the General Convergence Principle which states

Theorem ([2, Theorem I.5.1])

Let u_ϵ be the solution of a differential equation and u_i^h a difference approximation. Let $p > 0$ and C_1, C_2 be positive constants independent of h. Then, for all $i \geq 0$, all $0 < h < h_0$, and all $\epsilon > 0$

$$|u_\epsilon(x_i) - u_i^h| \leq C_1 h^p$$

iff

(i) $\lim_{h\to 0}|u_\epsilon(x_i) - u_i^h| = 0$

(ii) $|u_i^{2h} - u_i^h| \leq C_2 h^p$.

Furthermore C_1 is independent of ϵ iff C_2 is.

Essentially the method involves calculating the quantity given in (ii) which we shall call the double mesh error $e_{d\epsilon}^h$ and determining a rate of convergence

$$p_{d\epsilon}^h = (\ln e_{d\epsilon}^{2h} - \ln e_{d\epsilon}^h)/\ln 2) \tag{3}$$

where

$$e_{d\epsilon}^h = \max_{0\leq i\leq N} |u_i^{2h} - u_i^h|, \quad h = [x_L - x_R]/N. \tag{4}$$

The experimental uniform rate of convergence is then determined as

$$p = \min_\epsilon p_{d\epsilon} \quad \text{where} \quad p_{d\epsilon} = \text{average}_h p_{d\epsilon}^h. \tag{5}$$

Doolan, Miller and Schilders remark that the choice of the range of h values permissible is limited, since if the mesh is too coarse the solution of the difference scheme is not sufficiently representative of the solution of the differential equation to permit meaningful discussion of convergence, that is h is not *sufficiently small*, whereas if it is too fine then rounding error predominates. The method has the advantage that it requires no a priori knowledge of the nature of the solution of the equation and may be easily programmed to determine an experimental rate of convergence for a wide variety of problems.

Anomalies

In either of these methods however, great care must be taken in interpreting the table of values of $p_{\epsilon\epsilon}^h$ or $p_{d\epsilon}^h$ for the reasons which we will outline below. To simplify the arguments we shall consider only the rate of convergence, for a non-turning point problem, as considered in [6], that is :

$$\begin{aligned} \epsilon u'' + a(x)u' - b(x)u &= f(x), \quad 0 < x < 1 \\ u(0) = A, \qquad & u(1) = B. \end{aligned}$$

where $a(x) \geq \underline{a} > \alpha > 0$. The determination of the rate of convergence depends on the assumption that

$$e_{\epsilon\epsilon}^h < Ch^p \tag{6}$$

where C is independent of h and ϵ. This is not necessarily the strongest bound available and in fact the following one, (*cf.* [6]) , is a more accurate estimate :

$$e_{\epsilon\epsilon}^h < C(\frac{h^2}{h+\epsilon} + \frac{h}{\epsilon}e^{-\alpha h/\epsilon}) = Ch(\frac{\rho}{1+\rho} + \rho e^{-\alpha\rho}). \tag{7}$$

where $\rho = h/\epsilon$ and $\alpha > 0$. So more accurately

$$e_{\epsilon\epsilon}^h < C(\rho, \alpha)h.$$

which is not inconsistent with (6) since by considering the limits as $\rho \to 0$ and $\rho \to \infty$, for α fixed , we can see that $C(\rho, \alpha)$ is bounded. Let us assume that equality holds in (8) and determine the rate of convergence $p_{\epsilon\epsilon}^h$. Thus

$$p_\epsilon^\rho \equiv p_{\epsilon\epsilon}^h(\rho) = \ln\left[4\frac{(2\rho+1)^{-1} + e^{-2\alpha\rho}}{(\rho+1)^{-1} + e^{-\alpha\rho}}\right]/\ln 2) \tag{8}$$

and considering this for $\epsilon \to \infty (\rho \to 0)$ and $\epsilon \to 0 (\rho \to \infty)$ we get

$$\lim_{\rho\to 0} p_\epsilon^\rho = 2 \quad , \quad \lim_{\rho\to\infty} p_\epsilon^\rho = 1.$$

Thus, if we determine the rate of convergence p_ϵ^ρ, for fixed h, and ϵ varying from ∞ to 0, it will vary from 2 to 1 as expected. The assumption we implicitly make in evaluating computational orders of convergence is that p_ϵ^ρ is monotonic and hence that the minimum value is 1. This is not in fact the case as the function p_ϵ^ρ given by (9) may attain a minimum less than 1. This may be seen in Fig. 1 , which is a graph of p_ϵ^ρ for $\alpha = 0.25$. This lack of monotonicity is, in fact, most apparent when α is small. Similar results are also given in Fig. 1 for p_d^ρ.

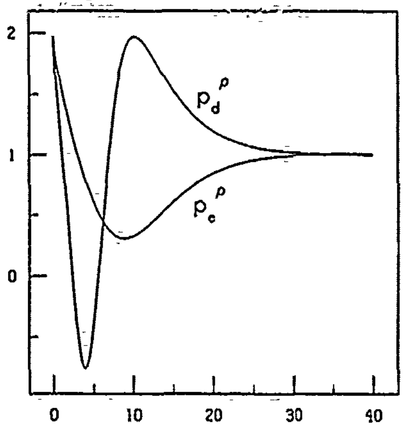


Figure 1. Double Mesh, p_d^ρ, and Exact, p_e^ρ, Rates of Convergence

ϵ	N	8	16	32	64	128	256	$p_{d\epsilon}$
1/	2	1.89	1.94	1.97	1.98	1.99	2.00	1.96
1/	4	1.81	1.89	1.94	1.97	1.98	1.99	1.93
1/	8	1.68	1.81	1.89	1.94	1.97	1.98	1.88
1/	16	1.48	1.68	1.81	1.89	1.94	1.97	1.79
1/	32	1.10	1.48	1.68	1.81	1.89	1.94	1.65
1/	64	0.33	1.10	1.48	1.68	1.81	1.89	1.38
1/	128	-0.75	0.03	1.10	1.48	1.468	1.81	0.94
1/	256	1.55	-0.75	0.33	1.10	1.48	1.68	0.90
1/	512	1.44	1.55	-0.75	0.33	1.10	1.48	0.86
1/	1024	1.02	1.44	1.55	-0.75	0.33	1.10	0.78
1/	2048	1.00	1.02	1.44	1.55	-0.75	0.33	0.76
1/	4096	1.00	1.00	1.02	1.44	1.55	-0.75	0.88
1/	8192	1.00	1.00	1.00	1.02	1.44	1.55	1.17
1/	16384	1.00	1.00	1.00	1.00	1.02	1.44	1.08
1/	32768	1.00	1.00	1.00	1.00	1.00	1.02	1.00
1/	65536	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1/	131072	1.00	1.00	1.00	1.00	1.00	1.00	1.00
p_d^h		1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 1: Double Mesh Rate of Convergence, p_d^ρ, $\alpha = 0.25$

the rate of convergence calculated using the Doolan, Miller, Schilders double-mesh method. In this case the actual rates are given in Table 1. It can be seen that serious problems can arise here in interpreting the table, particularly if we take the rate of convergence to be the minimum of $p_{d\epsilon}^h(\rho)$ over all ϵ, since this will be less than the actual rate of uniform convergence and for some problems may in fact be negative. This problem is significantly more noticeable for the rates calculated using the double mesh method than for those calculated using an exact or fine-mesh approximation. In the case $\alpha = 0.25$, for example, the minima are -0.75 and 0.32 respectively. This is what might be expected from comparison of the two curves in Fig. 1. The experimental rates of uniform convergence, for $\alpha = 0.25$, are $p_d = 0.76$ and $p_e = 0.80$ respectively, both of which are significantly lower than the true rate of 1.00. More examples of these phenomena are given in [5]. Reporting excessively low rates of uniform convergence is thus an expected feature of this method.

In view of these reservations we propose the following estimates for the rate of convergence

$$p_d^+ = \text{average}_h p_d^h \quad \text{and} \quad p_d^- = \min_h p_d^h$$

where

$$p_d^h = \ln(e_d^{2h} - e_d^h)/\ln(2)$$

and

$$e_d^h = \max_\epsilon e_{d\epsilon}^h = \max_\epsilon(\max_{0 \le i \le N} |u_i^{2h} - u_{2i}^h|).$$

It is clear that e_d^h is a function of h alone and thus we may expect p_d^h to be approximately a constant independent of h. This will lead to a better estimate for the uniform rate of convergence. In the case $\alpha = 0.25$, this is $p_d^+ = p_e^+ = p_d^- = p_e^- = 1.00$. The p_d^+-method has been used to determine the rate of convergence in [3, 4] and many later papers. We remark however that it does not give any additional information about the variations in behaviour of the scheme as $h \rightarrow 0$ or $\epsilon \rightarrow 0$. Therefore it is also useful to include tables of $p_{d\epsilon}^h$ to provide more precise details of this behaviour. This is particularly so for problems having more complicated boundary or interior layers.

We should remark at this point that there remain certain problems. In particular, the restriction that h was sufficiently large is crucial. If h becomes small, the most prevalent effect is for rounding errors to corrupt the results. However, if the calculations were done "exactly", so that rounding error were absent or negligible, then a more serious problem would arise. If we produce tables for arbitrarily small h, but only for *finitely* small ϵ, then most of the rates in the table will be for $h \ll \epsilon$. In this case, we are in the region where classical convergence theory applies and thus the rates will be greater than or equal to 1, for most schemes. These rates will dominate the table, and, if we use p_d^+ or p_e^+ as the calculated rate, cause even non-uniform schemes to be reported as uniformly convergent. This may be viewed as a consequence of the form of the tables, where, in this case in particular, the rate of convergence is a function of $\rho = h/\epsilon$. Thus the rates along the diagonals are equal. To get an accurate reflection of the rate of uniform convergence it is therefore necessary to extend the table at least as far in ϵ as in h. In the one dimensional cases, which we have tested, (*cf.* [3]), we extend the table until the errors, for given h, stabilizes, which occurs when one is solving, up to rounding error, the reduced equation. The finest mesh used in the calculations is $h = 1/4096$. In practice, using either double or fine mesh methods this has given acceptably accurate rates. In all cases the rate calculated using the fine mesh method proved higher. A more cautious approach might be to use p_d^- or p_e^-, which are less prone to this effect, although these will again report lower than actual rates of uniform convergence.

We remark that there are other circumstances in which these methods will report positive uniform convergence rates where, using the normal definition, the scheme would not be considered uniformly convergent. This is particularly true of the centered difference approximation to a self-adjoint problem and of two or higher dimensional problems exhibiting certain phenomena. These issues are considered further in [5].

References

[1] A. E. Berger, H. Han, and R. B. Kellogg, *A priori estimates and analysis of a numerical method for a turning point problem*, Math. Comp., 42 pp. 465-492, (1984).

[2] E.P. Doolan, J.J.H. Miller, W.H.A. Schilders, *Uniform numerical methods for problems with initial and boundary layers*, Boole Press, Dublin 1980.

[3] P.A. Farrell, *Uniformly convergent difference schemes for singularly perturbed turning and non-turning point problems* , Ph.D. thesis, Trinity College Dublin 1983.

[4] P.A. Farrell, *Sufficient conditions for the uniform convergence of a difference schemes for a singularly perturbed turning point problem*, SIAM J. Numer. Anal.,25 pp. 618-643, (1988).

[5] P.A. Farrell, A. Hegarty, *Some Comments on the Determination of the Order of Uniform Convergence*, Technical Report CS-91-03-02, Department of Mathematics and Computer Science, Kent State University, 1991.

[6] R.B. Kellogg, A. Tsan, *Analysis of some difference approximations for a singular perturbation problem without turning points*, Math. Comp., 32, pp. 1025-1039, (1978).

SPECIAL MESHES FOR TWO DIMENSIONAL ELLIPTIC SINGULAR PERTURBATION PROBLEMS

Alan F. Hegarty
Department of Mathematics
University of Limerick
Limerick
Ireland

John J.H. Miller
Department of Mathematics
Trinity College
Dublin 2
Ireland

Eugene O'Riordan
Department of Mathematics
Regional Technical College
Dundalk
Ireland

G.I. Shishkin
Ural Branch
Academy of Sciences
Sverdlovsk
U.S.S.R.

Abstract. In this paper, numerical examples are presented, which indicate that upwinded finite difference schemes on special meshes are numerically ε-uniformly convergent for the numerical solution of elliptic singular perturbation problems; for these examples it is also shown that upwinded finite difference schemes on uniform meshes behave unsatisfactorily. The numerical results given here validate the theoretical results obtained by the last author in [3] [4].

1. INTRODUCTION

In this paper, the following linear singular perturbation problem

$$Lu \equiv \varepsilon\Delta u + \vec{a}.\nabla u + a_0 u = f \quad \text{on} \quad \Omega = (0,1)\times(0,1) \qquad (1.1a)$$

$$u = g \quad \text{on} \quad \partial\Omega \qquad (1.1b)$$

where $\vec{a} = (a_1, a_2)$ and $0 < \varepsilon \leq 1$ is examined. Two cases are considered:

$$a_1 \geq \alpha_1, a_2 \geq \alpha_2 \quad \text{and} \quad \alpha_1 + \alpha_2 > 0 \qquad (1.1c)$$

$$a_1 \geq \alpha_1 > 0, \quad a_2 \equiv 0 \qquad (1.1d)$$

where α_1, α_2 are constants and a_1, a_2, a_0, f and g are smooth enough to ensure no interior or corner layers.

In each case, for small values of ε, boundary layers appear near some of the sides of Ω. In the case (1.1a,b,c) regular layers appear near $x = 0$ and $y = 0$ and in the case (1 1a,b,d) a regular layer appears near $x = 0$ and parabolic layers near the sides $y = 0$ and $y = 1$ which lie along characteristics of the reduced differential equation ($\varepsilon = 0$). For small values of ε, it is well known that classical numerical methods for (1.1) may produce spurious oscillations throughout the whole domain. Various stable upwind methods have been proposed to eliminate these oscillations. In §2, numerical examples demonstrate that the nodal errors for an upwinded method of this type (on a uniform mesh) depend not only on the number of mesh elements used but also on the value of ε. Numerical methods which converge independently of ε are usually said to be uniformly in ε convergent or ε-uniformly convergent. More precisely, a numerical method for solving (1.1) is ε-uniformly convergent of order p on the mesh $\omega_h = \{(x_i, y_j), i, j = 0, 1, \ldots, N\}$ if

$$\max_{\omega_h} \|u - z\| \leq CN^{-p}; \qquad (1.2)$$

where u is the solution of (1.1a.b,c) or (1.1a,b,d), z is the numerical approximation to u, C and $p > 0$ are independent of ε and N is the number of mesh elements used.

The following upwinded finite difference operator will always be used, to obtain a numerical approximation z to u

$$\varepsilon(\delta_{\bar{x}_i\hat{x}_i} + \delta_{\bar{y}_j\hat{y}_j})z + a_1(x_i, y_j)\delta_{x_i}z + a_2(x_i, y_j)\delta_{y_j}z$$
$$+ a_0(x_i, y_j)z = f(x_i, y_j) \quad (1.3)$$

where

$$\delta_{x_i}z \equiv (z(x_{i+1}, y_j) - z(x_i, y_j))/h_{i+1}$$
$$\delta_{\bar{x}_i}z \equiv (z(x_i, y_j) - z(x_{i-1}, y_j))/h_i$$
$$\delta_{\bar{x}_i\hat{x}_i}z \equiv (\delta_{x_i}z - \delta_{\bar{x}_i}z)/\bar{h}_i \text{ and} \qquad \bar{h}_i \equiv (h_{i+1} + h_i)/2,$$

on various meshes $\omega_h \equiv \{(x_i, y_j)\}$.

Shishkin [4] has proved that on a uniform mesh no ε-uniform finite difference scheme exists for a problem with a parabolic layer, such as (1.1a,b,d). In §3 a method which is ε-uniform for all types of layers is described. This method was first introduced by Shishkin in [3] and uses classical upwinded difference operators on a special piecewise-uniform mesh. Numerical results are presented for specific examples of (1.1a,b,c) and (1.1a,b,d).

2. Classical upwinding on a uniform mesh

In this section, the numerical performance of the upwinded difference scheme (1.3) on the uniform mesh

$$\omega_h^u \equiv \{(x_i, y_j) : x_i = ih, y_j = jh, i, j = 0, 1, \ldots, N\},$$

where $h = 1/N$ and N is the number of mesh elements used in both directions, is examined when applied to examples of (1.1a,b,c) and (1.1a,b,d). The problems are solved on a sequence of meshes, with N =8, 16, 32, 64, 128. The errors $|z(x_i, y_j) - u(x_i, y_j)|$ are approximated for successive values of ε on the four coarsest meshes by $e_\varepsilon^N(i,j) = |z^N(x_i, y_j) - z^{128}(x_i, y_j)|$, where the superscript indicates the number of mesh elements used. For each ε the maximum nodal error is approximated by

$$E_{\varepsilon,N} = \max_{i,j} e_\varepsilon^N(i,j).$$

Convergence rates for each ε and for $N = 8, 16, 32$, are estimated by $p_{\varepsilon,N}$ where

$$p_{\varepsilon,N} = \log_2\left(\frac{E_{\varepsilon,N}}{E_{\varepsilon,2N}}\right).$$

These estimated rates are given in tables 2.1 and 2.2 for the following problems:

Problem 2.1 $\varepsilon\Delta u + (2 + x^2 y)u_x + (1 + xy)u_y = 0$ on $(0,1)\times(0,1)$, with boundary conditions:

$$u(x,0) = 0; \quad u(x,1) = \begin{cases} 4x(1-x), & x > 1/2, \\ 1, & x \leq 1/2 \end{cases}$$

$$u(0,y) = 0; \quad u(1,y) = \begin{cases} 8(y - 2y^2), & y > 1/4, \\ 1, & y \leq 1/4 \end{cases}$$

Problem 2.2 $\varepsilon\Delta u + (1 + x^2 + y^2)u_x = 0$ on $(0,1)\times(0,1)$, with boundary conditions:$u(x,0) = x^3$; $u(x,1) = x^2$; $u(0,y) = 0$; $u(1,y) = 1$.

Problem 2.1 has regular layers near the sides $x = 0$ and $y = 0$; the estimated convergence rates are:

ε	N=8	N=16	N=32
1.0000000000	0.82	1.10	1.54
0.5000000000	0.92	1.13	1.54
0.2500000000	0.86	1.07	1.52
0.1250000000	0.84	0.97	1.46
0.0625000000	0.46	0.98	1.37
0.0312500000	-0.07	0.66	1.35
0.0156250000	-0.59	0.17	1.11
0.0078125000	-0.20	-0.40	0.69
0.0039062500	0.18	-0.24	0.08
0.0019531250	0.30	0.21	-0.13
0.0009765625	0.35	0.42	0.25
0.0004882813	0.38	0.55	0.66
0.0002441406	0.40	0.62	0.89
0.0001220703	0.40	0.67	1.06
0.0000610352	0.41	0.69	1.17

Problem 2.2 has a regular layer near the side $x = 0$ and parabolic layers near $y = 0$ and $y = 1$; the estimated convergence rates are:

ε	N=8	N=16	N=32
1.0000000000	1.72	1.52	1.38
0.2500000000	1.02	1.19	1.56
0.0625000000	0.49	1.01	1.43
0.0156250000	-0.39	0.12	0.96
0.0039062500	0.29	-0.39	0.17
0.0009765625	-0.90	0.67	-0.03
0.0002441406	-1.88	-0.69	0.96
0.0000610352	-1.94	-1.82	-0.27
0.0000152588	-1.95	-1.96	-1.50
0.0000038147	-1.95	-1.97	-1.86
0.0000009537	-1.95	-1.97	-1.95

In fact, it is well known that upwinding on a uniform mesh is not uniformly in ε convergent; however, in neither case is an estimate of the uniform convergence rate calculated because the maximum value of N is not large enough to make such estimates meaningful.

3. Numerical results on special meshes.

For the problem (1.1a,b,c), define the special mesh $\omega^*_{h,1}$ by

$$\omega^*_{h,1} \equiv \{(x^*_i, y^*_j) : 0 \leq i,j \leq N\}$$

where $x^*_i = ih_1$, for $0 \leq i \leq N/2$,

$$x^*_i = \sigma_x + (i - (N/2))h_2, \quad \text{for} \quad N/2 \leq i \leq N,$$

$$h_1 = 2\sigma_x/N, \quad \text{and} \quad h_2 = 2(1 - \sigma_x)/N.$$

The points $\{y^*_j\}$ are defined analagously The transition point σ_x is chosen to depend on both the layer width and the number of mesh elements used.

$$\sigma_x \equiv \min\{1/2, C_1\varepsilon \ln N\}, \quad C_1 > \alpha_1$$

The transition point σ_y is defined analagously. Using the upwinded difference operator (1.3) on this special mesh, we obtain the following estimates of convergence rates for problem 2.1:

ε	N=8	N=16	N=32
1.0000000000	0.81	1.10	1.53
0.5000000000	0.92	1.13	1.53
0.2500000000	0.86	1.07	1.51
0.1250000000	0.66	0.84	1.22
0.0625000000	0.90	1.01	1.43
0.0312500000	0.81	1.02	1.43
0.0156250000	0.77	0.99	1.40
0.0078125000	0.68	0.97	1.41
0.0039062500	0.60	0.91	1.38
0.0019531250	0.58	0.88	1.35
0.0009765625	0.57	0.86	1.34
0.0004882813	0.56	0.86	1.34
0.0002441406	0.56	0.85	1.34
0.0001220703	0.56	0.85	1.34
0.0000610352	0.56	0.85	1.33

For the problem (1.1a,b,d), define the special mesh $\omega^*_{h,2}$ by

$$\omega^*_{h,2} \equiv \{(x^*_i, y^*_j) : 0 \leq i,j \leq N\}$$

where $x_i = 2i\sigma_x/N, \quad i = 0, 1, \ldots, N/2,$

$$x_i = \sigma_x + 2(i - N/2)(1 - \sigma_x)/N, \quad i = N/2, \ldots, N,$$

$$y_i = 4i\sigma_y/N, \quad i = 0, 1, \ldots, N/4,$$

$$y_i = \sigma_y + 2(i - N/4)(1 - 2\sigma_y)/N, \quad i = N/4, \ldots, 3N/4,$$

$$y_i = (1 - \sigma_y) + 4(i - 3N/4)(\sigma_y)/N, \quad i = 3N/4, \ldots, N,$$

where $\sigma_x \equiv \min\{\frac{1}{2}, \varepsilon \ln N\}$, and $\sigma_y \equiv \min\{\frac{1}{4}, \sqrt{\varepsilon} \ln N\}$.

Using (1.3) on this mesh, we obtain the following convergence rate estimates for problem 2.2:

ε	N=8	N=16	N=32
1.0000000000	1.71	1.53	1.38
0.2500000000	1.02	1.19	1.56
0.0625000000	0.92	1.05	1.40
0.0156250000	0.91	1.07	1.43
0.0039062500	0.77	1.06	1.47
0.0009765625	0.78	1.01	1.48
0.0002441406	0.81	0.98	1.46
0.0000610352	0.82	0.97	1.44
0.0000152588	0.83	0.97	1.43
0.0000038147	0.83	0.97	1.43
0.0000009537	0.83	0.97	1.43

While, again, no estimate of the uniform convergence rate is given for either problem, the rates for each ε and h are evidently far superior to those obtained using a uniform mesh. A fuller discussion of the numerical experiments outlined above can be found in [1],[2].

References

1. Hegarty, A.F., Miller, J.J.H., O'Riordan, E. and Shishkin, G.I., Numerical methods for solving singularly perturbed problems from an engineering viewpoint, (to appear).

2. Hegarty, A.F., Miller, J.J.H., O'Riordan, E. and Shishkin, G.I., Special meshes for finite difference approximations to an advection-diffusion equation with parabolic layers, (to appear).

3. Shishkin G.I., Grid approximation of singularly perturbed parabolic equations with internal layers, *Sov. J. Numer. Anal. Math Modelling*, v.3, n.5, 1988, pp.393-407.

4. Shishkin, G.I. Approximation of the solution to singularly perturbed boundary value problem with parabolic layers, *J. Vychisl. Mat. i Mat. Fis.*, 29, No. 7, 1989, pp.963-977.

CONSTRUCTION AND ANALYSIS OF PETROV-GALERKIN APPROXIMATIONS FOR CONVECTION-DOMINATED FLOWS

B.W.SCOTNEY

Department of Mathematics, University of Ulster
Cromore Road, Coleraine, BT52 1SA.

Abstract - Error estimates are shown for Galerkin and Petrov-Galerkin approximations to a singularly perturbed problem in one dimension. An optimal Petrov-Galerkin formulation is presented for variable coefficient problems, and it is used to analyse the common practice of approximating variable coefficient problems by the use of locally constant methods.

1. INTRODUCTION

We consider the model problem

$$-au''(x) + b(x)u'(x) = f(x) \quad , \; x \in (0,1)$$
$$u(0) = g_L \;,\; u(1) = g_R \qquad (1.1)$$

where $a > 0$, $b(x) \in H^1$ with $b(x) > 0$ and $f(x) \in L_2$.

For the case when b is a positive constant Hemker (1977) developed a Petrov-Galerkin formulation with a piecewise linear trial space which generates a nodally exact solution. Here we develop an optimal Petrov-Galerkin approximation with a piecewise linear trial space for the general variable velocity problem (1.1).

2. WEAK FORMULATION AND ERROR ESTIMATES

The weak formulation of problem (1.1) is to find $u \in H^1_E$ such that

$$B(u,v) = \langle f,v\rangle \quad \forall\, v \in H^1_{E_0} \qquad (2.1)$$

where $B(\cdot,\cdot)$ is the bilinear form

$$B(w_1,w_2) = \langle w_1', \varepsilon w_2' + bw_2\rangle$$

For a piecewise linear trial space $S^h \subset H^1$ and $S^h_{E_o} = S^h \cap H^1_{E_o}$, the Galerkin approximation is to find $U \in S^h_E$ such that

$$B(U,V) = \langle f,V\rangle \quad \forall\, V \in S^h_{E_o} \qquad (2.2)$$

If we denote by $\|w\|_{T_1}$ the norm on $H^1_{E_o}$ defined by

$$\|w\|^2_{T_1} = \langle w',w'\rangle \qquad (2.3)$$

the Lax-Milgram Theorem (see, for example, Ciarlet (1978)), together with a Friedrichs-Poincaré inequality yields the following error estimate for the Galerkin approximation U:

$$\|u-U\|_{T_1} \le (1 + b/\pi a) \inf_{V\in S^h_E} \|u-V\|_{T_1} \qquad (2.4)$$

where $b = \max\limits_{x\in(0,1)} |b(x)|$.

If an Aubin-Nitsche duality argument (see Aubin (1972)) is used, an improved estimate can be achieved of the form

$$\|u-U\|_{T_1} \le (1 + bh/\pi a) \inf_{V\in S^h_E} \|u-V\|_{T_1} \qquad (2.5)$$

In Scotney (1985) the optimal estimate

$$\|u-U\|_{T_1} \le (1 + (bh/a)^2/12)^{1/2} \inf_{V\in S^h_E} \|u-V\|_{T_1} \qquad (2.6)$$

is obtained and shown to be the sharpest attainable bound.

The estimates (2.5) and (2.6) exhibit the importance of the mesh Péclet number bh/a in the degree to which optimality of the Galerkin approximation is lost as the differential operator loses self-adjointness.

Petrov-Galerkin methods may be formulated for singular perturbation problems with a view to recapturing the optimal approximation properties enjoyed by the Galerkin approximation for self-adjoint problems. A test space $T^h \subset H^1$ other than S^h is employed. Setting $T^h_{E_o} = T^h \cap H^1_{E_o}$, the Galerkin system (2.2) is replaced by the problem of finding $U \in S^h_E$ such that

$$B(U,V) = \langle f,V\rangle \quad \forall\, V \in T^h_{E_o} \qquad (2.7)$$

Morton (1982) shows how to achieve bounds sharper than those obtained directly from the Lax-Milgram Theorem, such as (2.4). He identifies the crucial requirements to establish optimal error estimates. By the Riesz Representation Theorem (see, for example, Adams (1975)), there exists a map $R: H^1_{E_o} \to H^1_{E_o}$ such that

$$B(v,w) = \langle v',(Rw)'\rangle \quad \forall\, v,w \in H^1_{E_o} \qquad (2.8)$$

If the constant Δ is defined by

$$\Delta = \sup_{V\in S^h_{E_o}} \inf_{W\in T^h_{E_o}} \frac{\|V-RW\|_{T_1}}{\|V\|_{T_1}} \qquad (2.9)$$

and U is the Petrov-Galerkin solution to problem (2.7) the following estimate holds:

$$\|u-U\|_{T_1} \le (1 - \Delta^2)^{-1/2} \inf_{V\in S^h_E} \|u-V\|_{T_1} \qquad (2.10)$$

This is the sharpest possible estimate since there will exist a function $f \in L_2$ for which (2.10) is an equality. (2.8), (2.9) and (2.10) provide the basis for Section 4.

3. PETROV-GALERKIN METHODS FOR CONSTANT COEFFICIENT PROBLEMS

We consider a uniform discretisation with nodes $x_j = jh$, $j = 0,\dots,N$, and $Nh = 1$.

For the problem (1.1) when $b > 0$ is a constant the test space $T^h_{E_o}$ proposed by Hemker (1977) has a basis $\{\psi^H_j,\ j = 1,\dots N-1\}$ with $\psi^H_j(x) =$

$$\begin{array}{ll} (e^{-\beta(x - x_{j-1})/h} - 1)/(e^{-\beta} - 1), & x_{j-1}<x\le x_j \\ (e^{-\beta(x - x_j)/h})/(e^{-\beta} - 1) & , x_j<x\le x_{j+1} \quad (3.1) \\ 0 & , \text{otherwise} \end{array}$$

where $\beta = bh/a$. The solution of (2.7) thus generated is nodally exact. It is straightforward to show that the best fit $U^* \varepsilon S_E^h$ to $u \varepsilon H_E^1$ in the norm $\|\cdot\|_{T_1}$ is also nodally exact, and hence that $U = U^*$.

Numerous Petrov-Galerkin methods have been proposed for problem (1.1) which generate the same finite difference operator as the test space defined by (3.1), namely that of Allen & Southwell (1955) - see, for example, Heinrich & Zienkiewicz (1979). Such methods are not optimal since they differ in their treatment of the source term f. However Scotney (1985) uses the estimate (2.10) of Morton (1982) to show that many are near-optimal. Griffiths & Lorenz (1978) have analysed these methods by direct use of the Lax-Milgram Theorem.

4. RIESZ REPRESENTATION

In the variable coefficient problem (1.1), from the defining relation (2.8) we may deduce that

$$(Rw)(x) = w(x) + \int_0^x (b(t)/a)w(t)dt - x\alpha(w) \qquad \forall \; w \; \varepsilon \; H_{E_0}^1 \qquad (4.1)$$

$$\text{where} \quad \alpha(w) = \int_0^1 (b(t)/a)w(t)dt \qquad (4.2)$$

From (2.9) and (2.10) it is clear that if $RT_{E_0}^h = S_{E_0}^h$ then the Petrov-Galerkin solution is optimal. We therefore aim to construct a basis $\{\psi_j, \; j = 1,\ldots,N-1\}$ such that $R\psi_j \; \varepsilon \; S_{E_0}^h$ with the additional property that the support of ψ_j is restricted to the support of the piecewise linear trial space basis function ϕ_j, namely (x_{j-1}, x_{j+1}). The key to the localisation is to set $R\psi_j$ to be of the same form as $R\psi_j^H$. In particular let

$$(R\psi_j)(x) = \psi_j(x) + \int_0^x (b(t)/a)\psi_j(t)dt - x\alpha_j \qquad \forall \; w \; \varepsilon \; H_{E_0}^1 \qquad (4.3)$$

$$\text{where} \quad \alpha_j = \int_0^1 (b(t)/a)\psi_j(t)dt \qquad (4.4)$$

$$\text{Writing} \quad g(x) = (R\psi_j)(x) + \alpha_j x$$

$$= \psi_j(x) + \int_0^x (b(t)/a)\psi_j(t)dt - x\alpha_j \qquad (4.5)$$

we obtain

$$\psi_j(x) = e^{-\int^x (b(t)/a)dt} \int_0^x g'(s)e^{\int^s (b(t)/a)dt}ds \qquad (4.6)$$

From Scotney (1985) we take the required form for g(x) as below:

$$\begin{array}{ll} 0 & , \quad 0<x\leq x_{j-1} \\ (T_j/h)(x - x_j) & , \; x_{j-1}<x\leq x_j \qquad (4.7) \\ T_j + (\alpha_j - T_j)(x - x_j)/h & , \; x_j<x\leq x_{j+1} \\ \alpha_j & , \; x_{j+1}<x \end{array}$$

The explicit form of $\psi_j(x)$ may be obtained by substituting (4.7) into (4.6), satisfying the requirement for local support and normalisation. Restricting the support of ψ_j to that of ϕ_j requires us to set $\psi_j(x) = 0$ for $x_{j+1}<x$,

$$\text{yielding} \quad T_j = \frac{\alpha_j \; \gamma_j}{(\gamma_j - \gamma_{j-1})} \qquad (4.8)$$

$$\text{where} \quad \gamma_{k-1} = \int_{x_{k-1}}^{x_k} e^{B(s)}ds \; , \; k = 1,\ldots,N \qquad (4.9)$$

$$\text{and} \quad B(x) = \int^x (b(t)/a)dt \qquad (4.10)$$

Normalising by setting $\psi_j(x_j) = 1$ gives

$$\alpha_j = \frac{h(\gamma_j - \gamma_{j-1})e^{B(x_j)}}{\gamma_{j-1}\gamma_j} \qquad (4.11)$$

and hence we obtain $\psi_j(x) =$

$$\begin{array}{ll} (e^{[B(x_j) - B(x)]}/\gamma_{j-1}) \int_{x_{j-1}}^x e^{B(s)}ds, & x_{j-1}<x\leq x_j \\ & (4.12) \\ (e^{[B(x_j) - B(x)]}/\gamma_j) \int_x^{x_{j+1}} e^{B(s)}ds, & x_j<x\leq x_{j+1} \\ 0 & ,\text{otherwise} \end{array}$$

It is straightforward to check that if $b(x) = b > 0$, a constant, (4.12) reduces to the test function ψ_j^H of Hemker (1977) as in (3.1).

If the test space generated by (4.12) is used in the Petrov-Galerkin formulation (2.7) it can be shown that the system of linear equations generated is of the form

$$\frac{a\alpha_j}{h(\gamma_j - \gamma_{j-1})} \left(-\gamma_j \; U_{j-1} + (\gamma_j + \gamma_{j-1})U_j - \gamma_{j-1} \; U_{j+1} \right) = \langle f, \psi_j \rangle \; j = 1,\ldots N-1 \quad (4.13)$$

The significance of (4.13) for the variable coefficient problem (1.1) is analagous to that of the Allen & Southwell (1955) difference operator for the constant coefficient problem.

5. LOCALLY CONSTANT METHODS FOR VARIABLE COEFFICIENT PROBLEMS

It has been commonplace for many authors to generalise the use of Hemker's test space (or its near-optimal counterparts) to variable coefficient problems by selecting the value of b in ψ_j^H to be determined locally at each node or in each element (e.g. $b = b(x_j)$ for $x \; \varepsilon \; (x_{j-1}, x_{j+1})$, or $b = (b(x_{j-1}) + b(x_j))/2$ for $x \; \varepsilon \; (x_{j-1}, x_j)$). Only with the availability of $\psi_j(x)$ as in (4.12) is it possible to properly analyse this practice.

Consider a modified form of problem (2.1) in which the velocity field $\underline{b}(x)$ is replaced by a piecewise constant field $\overline{b}(x)$ defined locally on each element: find $u \; \varepsilon \; H_E^1$ such that

$$\overline{B}(u,v) = \langle f,v\rangle \quad \forall\, v \in H^1_{E_o} \tag{5.1}$$

where $\overline{b}(x) = \overline{b}_j = (b(x_j) + b(x_{j+1}))/2$

for $x \in (x_j, x_{j+1})$,

$j = 0,\ldots N-1$, and $\overline{B}(\cdot,\cdot)$ is $B(\cdot,\cdot)$ with $b(x)$ replaced by $\overline{b}(x)$.

If we denote by $\overline{\psi}_j$ the test function ψ^H_j of Hemker with $b(x)$ replaced by $\overline{b}(x)$, we may consider the modified Petrov-Galerkin formulation: find $\overline{U} \in S^h_E$ such that

$$\overline{B}(\overline{U},\overline{\psi}_j) = \langle f,\overline{\psi}_j\rangle \quad j = 1,\ldots,N-1 \tag{5.2}$$

The formulation (5.2) is an obvious generalisation to the variable coefficient problem (1.1) of the use of Hemker's constant coefficient test space described in (3.1).

By substituting $\overline{b}(x)$ for $b(x)$ in (4.12), we can show that the optimal test space for problem (5.1) is precisely the one given by $\overline{\psi}_j$ in (5.2). That is, the obvious generalisation of Hemker's formulation generates the optimal approximation to the solution of the modified problem (5.1).

Moreover, $\overline{\psi}_j(x)$ may be written in the form

$$\overline{\psi}_j(x) = \begin{cases} \phi_j(x) + \xi_{j-1}\sigma_{j-1}(x) & , x_j<x\leq x_{j-1} \\ \phi_j(x) - \xi_j\,\sigma_j(x) & , x_j<x\leq x_{j+1} \end{cases} \tag{5.3}$$

where $\qquad \sigma_j(x_j) = \sigma(x_{j+1}) = 0$

and
$$\int_{x_j}^{x_{j+1}} \sigma_j(x)\,dx = \frac{h}{2} \tag{5.4}$$

Then the system of linear equations generated by (5.2) may be written as

$$\begin{aligned} &(-a/h - (1 + \xi_{j-1})\overline{b}_{j-1}/2)\ U_{j-1} \\ &+ (2a/h + ((1 + \xi_{j-1})\overline{b}_{j-1} - (1 - \xi_j)\overline{b}_j)/2)\ U_j \\ &+(-a/h + (1 - \xi_j)\overline{b}_j/2)\ U_{j+1} = \langle f,\overline{\psi}_j\rangle \\ &\qquad\qquad j = 1,\ldots,N-1 \end{aligned} \tag{5.5}$$

Since (5.4) implies no knowledge of the functional form of $\sigma_j(x)$, any of the conforming Petrov-Galerkin test spaces used for the constant coefficient problem can be modified to fit (5.3) and hence used to generate the left hand side of (5.5).

REFERENCES

Adams, R.A., 1975. Academic Press, New York.

Allen, D.N. de G. & Southwell, R.V., 1955. Quart. J. Mech. Appl. Math., 8, pp 129-145.

Aubin, J.P., 1972. John Wiley & Sons, New York.

Ciarlet, P.G., 1978. North Holland, Amsterdam.

Griffiths, D.F. & Lorenz, J., 1978. Comp. Meth. Appl. Mech. & Eng., 14, pp 39 - 64.

Heinrich, J.C. & Zienkiewicz, O.C., 1979. A.M.D. - Vol 34, A.S.M.E., New York, pp 105-136.

Hemker, P.W., 1977. Thesis, Mathematisch Centrum, Amsterdam.

Morton, K.W., 1982. Lecture Notes in Mathematics, 965, Springer-Verlag, Berlin, pp 113 - 148.

Scotney, B.W., 1985. Ph.D. Thesis, University of Reading.

EXPONENTIALLY FITTED BOX METHODS AND THEIR APPLICATION TO SEMICONDUCTOR DEVICE SIMULATION

W.H.A. Schilders
Philips Research Laboratories
Applied Mathematics Group
Building WAY, Room 2.09
PO Box 218, 5600 MD, Eindhoven (NL)

ABSTRACT

The system of differential equations describing the behaviour of semiconductor devices is singularly perturbed, and therefore requires special discretisation techniques. To this end, an exponentially fitted box method has been developed, which is known as the Scharfetter-Gummel method In this paper, we will discuss this discretisation technique, which is applicable to n-dimensional problems. Application of this technique is not restricted to semiconductor device problems, it is suitable for any singularly perturbed problem in divergence form It is conjectured that this method yields uniform error estimates for arbitrary polygonal meshes. It is also shown that the 1-d scheme is essentially the same as Il'ins scheme, thus paving the way for uniform error estimates on the solution of the semiconductor problem. On the other hand, the way the scheme has been constructed is different from the construction of Il'ins scheme, and therefore this may be of interest to singular perturbationists. Finally extensions of the scheme to a special case are discussed, where exponentially fitted methods for small systems of equations can be used. Again it is suspected that uniform error estimates may be obtained.

1. SEMICONDUCTOR DEVICE SIMULATION

The differential equations describing the behaviour of semiconductor devices are derived from the Maxwell equations and from Boltzmann's equation. An excellent account of this can be found in [10,12]. In order to adequately describe the behaviour of semiconductor devices, another charged particle is introduced, namely the positively charged hole. Thus, currents are not only caused by moving electrons, but also by moving holes.

We will restrict ourselves to the following system of equations:

$$\nabla \cdot (\varepsilon \mathbf{E}) = q(p - n + D) \qquad (1)$$

$$\nabla \cdot \mathbf{J}_p = -qR \qquad (2)$$

$$\nabla \cdot \mathbf{J}_n = qR \qquad (3)$$

where the electric field $\mathbf{E}$, the hole current density $\mathbf{J}_p$ and the electron current density $\mathbf{J}_n$ are given by

$$\mathbf{E} = -\nabla\psi \qquad (4)$$

$$\mathbf{J}_p = -q\mu_p(U_T\nabla p - p\mathbf{E}) \qquad (5)$$

$$\mathbf{J}_n = q\mu_n(U_T\nabla n + n\mathbf{E}) \qquad (6)$$

Suitable boundary conditions are applied. The variables in this system are the electric potential ψ, the hole concentration p and the electron concentration n.

For the application of exponentially fitted methods, it is sometimes convenient to rewrite the system (1)-(6) into other variables. First of all, the electric potential and the carrier concentrations differ very much in size. Therefore, it is convenient to introduce the quasi-Fermilevels ϕ_p and ϕ_n defined by the relations

$$p = n_{int}e^{(\phi_p-\psi)/U_T} \qquad (7)$$

$$n = n_{int}e^{(\psi-\phi_n)/U_T} \qquad (8)$$

where n_{int} is the intrinsic carrier concentration. In this case, the equations (5)-(6) become

$$\mathbf{J}_p = -q\mu_p n_{int}e^{(\phi_p-\psi)/U_T}\nabla\phi_p \qquad (9)$$

$$\mathbf{J}_n = -q\mu_n n_{int}e^{(\psi-\phi_n)/U_T}\nabla\phi_n \qquad (10)$$

A third choice of variables can be motivated by remarking that the system of equations can be put into self-adjoint form, which is convenient in designing suitable discretisation schemes. This can be achieved by applying a Liouville transformation. The result is the formulation of the semiconductor problem in the set of variables ψ, Φ_p and Φ_n, the latter two being termed the Slotboom variables (cf. [11]). These are defined by the relationships

$$p = n_{int}e^{-\psi/U_T}\Phi_p \qquad (11)$$

$$n = n_{int}e^{\psi/U_T}\Phi_n \qquad (12)$$

In this case, equations (5)-(6) read.

$$\mathbf{J}_p = -q\mu_p U_T n_{int}e^{-\psi/U_T}\nabla\Phi_p \qquad (13)$$

$$\mathbf{J}_n = q\mu_n U_T n_{int}e^{\psi/U_T}\nabla\Phi_n \qquad (14)$$

2. SINGULARLY PERTURBED CHARACTER

First we consider equation (1), which is often referred to as Poisson's equation. We scale the equation and the variables as described in [7,8]. If we then rewrite the Poisson equation in terms of the scaled quantities, we obtain

$$\lambda^2\nabla \cdot \nabla\psi = n - p - D$$

where the right hand side is of the order of unity. The parameter λ is rather small. Typically, its value is of the order of magnitude $10^{-3} - 10^{-5}$. Thus, from a mathematical point of view, the Poisson equation is singularly perturbed.

The singularly perturbed character of Poisson's equation is not of too much interest, since it is a self-adjoint equation for which the application of standard difference schemes does not lead to erroneous solutions (although accuracy could be an issue). Much more interesting is that, because of the size of the parameter λ, the entire system (1)-(6) is singularly perturbed. For more details on the demonstration of this fact we refer the reader to [7]. This book also contains asymptotic expansions for the solution, which is important in view of the necessary conditions for a difference scheme to possess the property of uniform convergence (cf [illegible]). For the discussion in this paper it will suffice to remark that, because of the behaviour of the electric field, the equations (5) and (6) can be considered as singularly perturbed first order equations for p and n, respectively. This is an important observation, since it explains why standard difference schemes can not be applied to the discretisation of these equations.

3. EXPONENTIAL FITTING

As has been discussed in the foregoing section, (5) and (6) can be considered as singularly perturbed first order differential equations for p and n, respectively. Application of the standard central difference scheme on a mesh $\{x_0, \ldots, x_N\}$ leads to the following expression for the hole current density in the interval $[x_i, x_{i+1}]$ (denoted by $J_{p,i+1/2}$):

$$J_{p,i+1/2} = -q\mu_p(U_T\frac{p_{i+1} - p_i}{x_{i+1} - x_i} - \frac{p_{i+1} + p_i}{2}E_{i+1/2}) \qquad (15)$$

where

$$E_{i+1/2} = -\frac{\psi_{i+1} - \psi_i}{x_{i+1} - x_i} \qquad (16)$$

A similar expression is obtained for the electron current density. These expressions are then used in the discrete versions of equations (1)-(5) which, for (2), reads:

$$J_{p,i+1/2} - J_{p,i-1/2} = -q(x_{i+1/2} - x_{i-1/2})R_i \qquad (17)$$

It is wellknown that this scheme yields non-stable solutions. In fact, closer investigation shows that the concentrations may become negative, meaning that no solution in terms of the variables ψ, ϕ_p, ϕ_n exists. Thus, the discrete scheme does not guarantee existence of solutions.
Applying exponential fitting techniques to the discretisation yields

$$J_{p,i+1/2} = -q\mu_p(\sigma_i U_T\frac{p_{i+1} - p_i}{x_{i+1} - x_i} - \frac{p_{i+1} + p_i}{2}E_{i+1/2}) \qquad (18)$$

with

$$\sigma_i = \frac{E_{i+1/2}(x_{i+1} - x_i)}{2U_T}\coth(\frac{E_{i+1/2}(x_{i+1} - x_i)}{2U_T}) \qquad (19)$$

which is exactly the Il'in fitting factor (cf. [6]).
This scheme has first been described by Scharfetter and Gummel (cf. [9]), and therefore the resulting difference scheme is known to device modellers as the Scharfetter-Gummel scheme. As we remarked in [illegible] above, it is exactly the same as Il'ins scheme. A coincidence is that both schemes were developed in 1969!
Because the above method is the same as Il'ins scheme, (uniform) error estimates for the latter can be carried over directly to the semiconductor problem. Then we obtain:

Theorem 1
We have the following error estimate for the discrete solution $\{p_i, i = 0, ..., N\}$:

$$\frac{|p_i - p(x_i)|}{p(x_i)} \leq Ch$$

where h is the maximum mesh spacing and C is a constant independent of h and the coefficients in the equation.

□

Since the estimate here concerns the relative error, the result can also be rewritten as an error estimate for the quasi-Fermilevel ϕ_p.
The scheme in (18) has been derived in the above by applying exponential fitting techniques. On the other hand, it is very interesting to see the relationship of this derivation with the way it is normally derived in the area of semiconductor device modelling. To this end, we again consider the homogeneous form of the second order differential equation obtained by combining (2) and (5). From (2) it then follows that J_p is constant; since this can be concluded for each interval separately, we will assume that J_p is piecewise constant. Using this and the expression for J_p in terms of the Slotboom variable Φ_p (cf (13)), we obtain:

$$\nabla\Phi_p = -\frac{J_p}{q\mu_p U_T n_{int}}e^{\psi/U_T}$$

$$\Rightarrow \Phi_{p,i+1} - \Phi_{p,i} = -\frac{J_p}{q\mu_p U_T n_{int}}\int_{x_i}^{x_{i+1}} e^{\psi/U_T}dx$$

and, assuming that ψ is linear in $[x_i, x_{i+1}]$:

$$\Phi_{p,i+1} - \Phi_{p,i} = -\frac{J_p}{q\mu_p U_T n_{int}}\frac{U_T}{\psi'}(e^{\psi_{i+1}/U_T} - e^{\psi_i/U_T})$$

or

$$\Phi_{p,i+1} - \Phi_{p,i} = -\frac{J_p}{q\mu_p U_T n_{int}}\frac{x_{i+1} - x_i}{\psi_{i+1}/U_T - \psi_i/U_T}(e^{\psi_{i+1}/U_T} - e^{\psi_i/U_T})$$

Rearranging this expression leads to the following expression for J_p in the interval $[x_i, x_{i+1}]$, which we again denote by $J_{p,i+1/2}$:

$$J_{p,i+1/2} = -q\mu_p U_T n_{int}\frac{\psi_{i+1}/U_T - \psi_i/U_T}{e^{\psi_{i+1}/U_T} - e^{\psi_i/U_T}}\frac{\Phi_{p,i+1} - \Phi_{p,i}}{x_{i+1} - x_i} \qquad (20)$$

Using the relations between the carrier concentrations and the Slotboom variables it is very easy to show that (20) is exactly the same as (18). On closer examination of (20) we see that, in fact, the discretisation of the rapidly varying coefficient e^{ψ/U_T} has been performed by using a harmonic average of this coefficient. In other words, for an approximation of this coefficient on the interval $[x_i, x_{i+1}]$ we have in fact taken

$$(e^{-\psi/U_T})_{[x_i,x_{i+1}]} \sim \frac{1}{\int_{x_i}^{x_{i+1}} e^{\psi/U_T}dx}$$

This technique, for second order differential equations in divergence form with rapidly varying coefficients, has also been described and analysed in [1]. Also, the so-called generalized finite element method of [2] works along similar ideas. Finally, the above technique of harmonic averaging appears in a natural way in the mixed finite element method when applied to semiconductor problems. For more details we refer the reader to several papers by Brezzi and co-workers (cf. [3,4]).

4. EXTENSION TO HIGHER DIMENSIONS

The box method is ideally suited for the discretisation of equations of the form

$$\nabla \cdot \mathbf{F} = f \qquad (21)$$

The starting point is a mesh, consisting of triangles, rectangles, quadrilaterals or other polygons. A so-called box B_i is constructed around each mesh point x_i, in such a way that the union of all boxes is the entire simulation domain and such that boxes do not overlap. The most common way of doing this is to construct B_i using the midperpendiculars of the mesh sides. Having completed this construction, we integrate (21) over each of the boxes:

$$\int_{B_i} \nabla \cdot \mathbf{F}dV = \int_{B_i} f dV$$

which, using Gauss' theorem, leads to

$$\int_{\partial B_i} \mathbf{F} \cdot \mathbf{n}\, dS = \int_{B_i} f dV \qquad (22)$$

The discretisation is now completed by approximating the integrals in this equation. A standard procedure is to use the lowest order quadrature-rules for both integrals. Thus, the right hand side of (22) is approximated by $\text{vol}(B_i)\, f_i$. The left hand side is approximated by

$$\sum_k l_k \mathbf{F} \cdot \mathbf{n}_k$$

where the $\mathbf{n}_k$ are the normals in the midpoints of the mesh sides and l_k are the lengths of the corresponding box sides.
Equations (1)-(3) can be discretised using the box method. The remaining problem is to evaluate the normal components of the electric field $\mathbf{E}$ and the current densities $\mathbf{J}_p$ and $\mathbf{J}_n$. Because of the construction of the boxes, however, this is rather straightforward. Namely, the normal components are exactly the components along the mesh sides. Thus, to obtain the desired quantities, we can just use the techniques outlined in the previous section. In other words: the one-dimensional arguments used to obtain exponentially fitted discretisations of the first order differential equations for J_p and J_n between two mesh points can be used here. For the electric field the situation is even simpler, because we do not need any fitting. In that case we just use expressions of the form (16).

Conclusion

The box scheme for equations (1)-(3), combined with the exponentially fitted schemes for equations (4)-(6), provide a suitable discretisation method for the semiconductor equations in 1-d, 2-d and 3-d.

□

In fact, we conjecture here that the proposed exponentially fitted box schemes yield uniform error estimates (this has been shown in 1-d and in the 2-d rectangular case).

5. EXPONENTIAL FITTING FOR SMALL SYSTEMS: THE CASE OF AVALANCHE GENERATION

In this section we generalise the Scharfetter-Gummel method to the case where the effect of avalanche generation is taken into account, using exponential fitting techniques for singularly perturbed 2 × 2 systems as described in [5]. This effect, which may lead to breakdown of the device, is accounted for in the equations by an extra recombination/generation term. Normally, these terms only depend on the carrier concentrations, and not on the current densities. Therefore, (2) and (3) are first order equations in J_p and J_n without any zero order terms. From a mathematical point of view this means that the current densities will vary rather smoothly. Remark that the assumption of piecewise constant current densities made in the derivation of the Scharfetter-Gummel scheme agrees with this.

When avalanche generation is taken into account, the situation changes completely. In this case, an extra term has to be included of the form

$$R_{II} = \frac{1}{q}(\alpha_p|J_p| + \alpha_n|J_n|) \tag{23}$$

Now we see that the first order differential equations do contain a zero order term, thus allowing the possibility of rapidly (namely. exponentially) varying current densities. It is clear that the assumption of piecewise linear current densities may not be adequate in this case.

As a starting point for our derivation we take the homogeneous form of equations (2) and (3), i.e. R is equal to the impact ionisation term given in (23). The equations read

$$J_p' = +\alpha_p|J_p| + \alpha_n|J_n| \tag{24}$$

$$J_n' = -\alpha_p|J_p| - \alpha_n|J_n| \tag{25}$$

which can be written as

$$\mathbf{J}' = A\mathbf{J} \tag{26}$$

where $\mathbf{J} = (J_p, J_n)^T$ and

$$A = \begin{pmatrix} +v_p\alpha_p & +v_n\alpha_n \\ -v_p\alpha_p & -v_n\alpha_n \end{pmatrix}$$

Here, $v_p = \text{sign}(J_p)$, $v_n = \text{sign}(J_n)$ are the 'directions' of J_p and J_n. Using equations (9)-(10) we see that $v_p = -\text{sign}(\phi_p')$, $v_n = -\text{sign}(\phi_n')$. Thus, (26) may be considered as a linear equation in J.

Now we are ready to discretise the problem. As usual, we take a mesh $\{x_0, ..., x_N\}$, the midpoint of the interval $[x_i, x_{i+1}]$ being denoted by $x_{i+1/2}$. We assume that A is constant on each of these intervals (notation: $A_{i+1/2}$). Multiplying (26) on the interval $[x_i, x_{i+1}]$ by $e^{-(x-x_{i+1/2})A_{i+1/2}}$, we obtain:

$$e^{(x-x_{i+1/2})A_{i+1/2}}[e^{-(x-x_{i+1/2})A_{i+1/2}}\mathbf{J}]' = 0$$

Thus, we may conclude that for solutions J of the continuous problem (26) we have that $e^{-(x-x_{i+1/2})A_{i+1/2}}\mathbf{J}$ is a constant vector

To proceed we now use equations (13) and (14). These can be written in the form

$$\mathbf{J} = D\Phi'$$

where $\Phi = (\Phi_p, \Phi_n)^T$ and

$$D = \begin{pmatrix} -q\mu_p U_T n_{int} e^{-\psi/U_T} & 0 \\ 0 & q\mu_n U_T n_{int} e^{\psi/U_T} \end{pmatrix}$$

Since $e^{-(x-x_{i+1/2})A_{i+1/2}}\mathbf{J}(x)$ is a constant vector, which we will denote by $\mathbf{J}_{i+1/2}$ (substitute $x = x_{i+1/2}$!), we have that

$$\Phi' = D^{-1}e^{(x-x_{i+1/2})A_{i+1/2}}\mathbf{J}_{i+1/2}$$

and it follows that

$$\Phi_{i+1} - \Phi_i = \{\int_{x_i}^{x_{i+1}} D^{-1}e^{(y-x_{i+1/2})A_{i+1/2}}dy\}\mathbf{J}_{i+1/2}$$

From this we obtain a discrete expression for the current densities on the interval $[x_i\ x_{i+1}]$:

$$\mathbf{J}(x) = e^{(x-x_{i+1/2})A_{i+1/2}}\mathbf{J}_{i+1/2} \tag{27}$$

where

$$\mathbf{J}_{i+1/2} = [\int_{x_i}^{x_{i+1}} D^{-1}e^{(y-x_{i+1/2})A_{i+1/2}}dy]^{-1}(\Phi_{i+1} - \Phi_i) \tag{28}$$

The final discretisation of (26) is then obtained by integrating over the box $[x_{i-1/2}, x_{i+1/2}]$:

$$\mathbf{J}_{i+1/2} - \mathbf{J}_{i-1/2} = \int_{x_{i-1/2}}^{x_i} A_{i-1/2}\mathbf{J}(y)dy + \int_{x_i}^{x_{i+1/2}} A_{i+1/2}\mathbf{J}(y)dy \tag{29}$$

Equation (29) can be expanded further by substituting the expressions (27) into the integrals. This leads to the following discretisation.

$$e^{(x_i-x_{i+1/2})A_{i+1/2}}\mathbf{J}_{i+1/2} - e^{(x_i-x_{i-1/2})A_{i-1/2}}\mathbf{J}_{i-1/2} = 0 \tag{30}$$

References

[1] O. Axelsson, *A generalized conjugate direction method and its application on a singular perturbation problem*, Numerical Analysis, G.A. Watson (ed.), LNiM, vol. 773, pp. 1-11 (1979)

[2] I. Babuska, J.E Osborn, *Generalized finite element methods. their performance and their relation to mixed methods*, SIAM J. Numer. Anal., vol 20, pp. 510-536 (1983)

[3] F. Brezzi, L.D. Marini, P. Pietra, *Mixed exponential fitting schemes for current continuity equations*, Proc. NASECODE VI Conf., Boole Press, Dublin (1989)

[4] F. Brezzi, L.D. Marini, P. Pietra, *Two-dimensional exponential fitting and applications to drift-diffusion models*, SIAM J. Numer. Anal. (to appear)

[5] E.P. Doolan, J.J.H. Miller, W.H.A. Schilders, *Uniform numerical methods for problems with initial and boundary layers*, Boole Press, Dublin (1980) (*also available in Russian and Chinese*)

[6] A.M. Il'in, *A differencing scheme for a differential equation with a small parameter affecting the highest derivative*, Math. Notes Acad. Sc. USSR, vol. 6, pp. 596-602 (1969)

[7] P.A. Markowich, *The stationary semiconductor equations*, Computational Microelectronics, S. Selberherr (ed.), Springer Verlag, Wien, New York (1986)

[8] S.J. Polak, C. den Heijer, W.H.A. Schilders, P. Markowich, *Semiconductor device modelling from the numerical point of view*, Int. J. Numer. Methd. Engng., vol. 24, pp. 763-838 (1987)

[9] D.L. Scharfetter, H.K. Gummel, *Large-signal analysis of a silicon Read diode oscillator*, IEEE Trans. Electron Devices, vol. ED-16, pp. 64-77 (1969)

[10] S. Selberherr, *Analysis and simulation of semiconductor devices*, Springer Verlag, Wien, New York (1984)

[11] J.W. Slotboom, *Computer-aided two-dimensional analysis of bipolar transistors*, IEEE Trans. Eelectron Devices, vol. ED-20, pp. 669-679 (1973)

[12] S.M. Sze, *Physics of semiconductor devices*, Wiley, New York (1969)

GRID APPROXIMATION OF BOUNDARY VALUE PROBLEMS FOR QUASILINEAR SINGULARLY PERTURBED ELLIPTIC EQUATIONS IN THE CASE OF FULL DEGENERATION

J.J.H. MILLER
School of Mathematics
Trinity College Dublin

and

GRIGORII I. SHISHKIN
Institute of Mathematics and Mechanics
Ural Branch of USSR Academy of Sciences
Sverdlovsk, USSR

Abstract A quasilinear singularly perturbed elliptic equation degenerating into an equation not containing derivatives is considered. Problems which appear when constructing numerical methods based, in particular, on fitting techniques are discussed. In the case of the Dirichlet problem on a strip, or on a domain with smooth curvilinear boundary, principles for the construction of uniformly (with respect to the parameter) convergent schemes are considered.

I. PROBLEM FORMULATION

On a strip $D = \{x : 0 < x_1 < d,\ |x_s| < \infty,\ s = 2, \cdots, n\}$ the following Dirichlet problem for a quasilinear equation is considered

$$L(u(x)) \equiv \varepsilon^2 L^1(u(x))u(x) - f(x, u(x)) = 0,$$
$$x \in D, u(x) = \varphi(x), x \in \Gamma. \qquad (1)$$

Here $L^1(v)$ is the second order elliptic operator

$$L^1(v) \equiv \sum_{s,k=1}^{n} a_{sk}(x,v)\partial^2/\partial x_s \partial x_k + \sum_{s=1}^{n} b_s(x,v)\partial/\partial x_s - c(x,v),$$

It is assumed that the coefficients of the operator L^1 and the functions f and φ are sufficiently smooth, and that for the function $f(x,u)$ the following condition is satisfied

$$(\partial/\partial u)f(x,u) \geq \alpha > 0, \quad (x,u) \in \overline{D} \times R^1.$$

The parameter ε can take any value in $(0,1]$. When ε tends to zero in a neighbourhood of the boundary Γ a boundary layer appears. This boundary layer is described by an ordinary differential equation. Derivatives of the solution along all directions out of a neighbourhood of the boundary layer and also derivatives along directions collinear with the boundary domain in the neighbourhood of the boundary layer are bounded uniformly with respect to the parameter. The numerical solution of such boundary-value problems is difficult even for linear equations and gives rise to the problem of constructing approximations on special grids, which converge uniformly with respect to the parameter (see, for example, [1]).

II. TYPICAL NUMERICAL PROBLEMS FOR NONLINEAR EQUATIONS

On a uniform grid it is impossible to construct a numerical solution of problem (1) converging uniformly with respect to the parameter. This is because, on the one hand problem (1) is nonlinear and, on the other hand the difference of the values of the solution to problem (1) on neighbouring nodes of the grid does not converge to zero uniformly with respect to the parameter (when the grid size tends to zero). For this reason it is necessary to construct difference schemes on grids condensing in the boundary layer.

III. THE BOUNDARY VALUE PROBLEM ON THE STRIP

First suppose that the coefficients of the operator L^1 do not depend on $u(x)$. Then the solution to problem (1) is the limit of a sequence of solutions to the boundary value problem

$$L^2\, u^{(k)}(x) \equiv (\varepsilon L^1 - \alpha)u^{(k)}(x) = f_\alpha(x, u^{(k-1)}(x)),$$
$$x \in D; \quad u^{(k)}(x) = \varphi(x), \quad x \in \Gamma \qquad (2)$$

Note that this equation is linear with respect to $u^{(k)}(x)$ and that $f_\alpha(x,u) = f(x,u) - \alpha u$. To solve problem (1) the iterative difference scheme

$$\Lambda^2\, z^{(k)}(x) = f\,(x, z^{k-1}(x)),$$
$$x \in D_h,\ z^{(k)}(x) = \varphi(x), x \in \Gamma_h \qquad (3)$$

aproximating problem (2) is used. Derivatives are approximated by classical finite differences on rectangular grids (see, for example, [2]), which are condensed in a neighbourhood of the boundary layer by a special rule (one such grid condensing rule is given in [3]). The conditions which guarantee uniform with respect to the parameter convergence of the solution to the solution of problem (3) (when $k \to \infty$ and the number of nodes increases) are indicated. A similar method is used when the coefficients of the operator L^1 depend on the solution of problem (1).

IV. DOMAINS WITH CURVILINEAR BOUNDARIES

In the case of domains with smooth curvilinear boundaries iterative schemes based on the alternating method of Schwartz are constructed. A neighbourhood of the boundary Γ is covered by a system of overlapping subdomains. In each subdomain the boundary Γ of the initial domain D is rectified by using a coordinate transformation. A difference scheme (in the new coordinate system) is constructed by employing results obtained for the boundary value problem on the strip. For boundary value problems on domains with curvilinear boundaries, conditions guaranteeing uniform with respect to the parameter convergence of the constructed difference scheme are deduced.

REFERENCES

1. Doolan E.P., Miller J.J.H., Schilders W.H.A. Uniform numerical methods for problems with initial and boundary layers. Dublin: Boole Press, 1980.

2. Samarsky A.A. Theory of difference schemes. Moscow: Nauka, 1977 (in Russian).

3. Shishkin G.I. Approximation of the solution of singularly perturbed boundary value problems with a corner boundary layer, Zh. Vyschisl.Mat.i Mat.Fis., 1987, T.27, N 9. P.1360-1374 (in Russian).

SPLITTING-TIME AND EXPONENTIAL FITTING-SPACE DISCRETIZATIONS FOR CONVECTION-DIFFUSION PROBLEMS.

C. CLAVERO, J.C. JORGE, F. LISBONA.
Departamento de Matemática Aplicada,
Universidad de Zaragoza,
Zaragoza (Spain).

Abstract. In this paper we develope a finite-difference method for time dependent convection-dominating convection-diffusion problems, modeled by the singularly perturbed parabolic equation

$$\frac{\partial u}{\partial t} - \varepsilon\Delta u + \vec{v}\vec{\nabla}u + ku = f, \text{ in } \Omega \times [0,T],$$

where $\Omega \subset I\!R^2, k \geq 0$ and ε is a (possibly small) positive constant.

This method may be viewed as a combination between Fractional Steps time semidiscretizations, and one-dimensional space discretizations of exponential fitting type. Making use of the consistency and the contractivity of the time integration process, and the special properties of the discretizations made, via exponential fitting, for the space differential operators at each time level, a (uniform in ε) convergence result is proven. Important advantages for computations are also obtained.

I. INTRODUCTION

Let Ω be a bounded domain in $I\!R^2$ with a piecewise smooth boundary Γ. We will deal with the numerical aproximation of the solution of time dependent, convection dominated, convection-diffusion problems defined by the equation

$$\frac{\partial u}{\partial t} - \varepsilon\Delta u + \vec{v}\vec{\nabla}u + ku = f, \text{ in } \Omega \times [0,T],$$

and the initial-boundary conditions

$$u(x,0) = u_0(x) \text{ in } \Omega$$
$$u(x,t) = g(x,t) \text{ in } \Gamma \times [0,T],$$

where $\vec{v} = (v_1, v_2)$ and k are smooth functions on $\bar{\Omega}$, with $\varepsilon > 0$ and such that it may occur $\varepsilon << |v|$. In general, the solution of this problem is not globally smooth (even for smooth data) but it may present rapid variations in certain narrow regions in Ω (layers).

Problems of this kind are found in the modelling of convection-conduction of heat and atmospheric transport of pollutants.

It is well-known that standard finite difference or finite element methods applied to convection dominated flows problems give unphisical oscilatory solutions with a reasonable mesh size; in these cases the maximum principle property is not preserved. The first remedy to avoid this drawback in finite differences was to introduce upwind differencing for the convective term. A related concept was adapted later to finite element methods.

These procedures generate spurious crosswind diffusion. More sophisticated techniques were developed with the idea of introducing an artificial viscosity term which acted only in the direction of the streamlines (streamline diffusion, Hughes & Brooks, Johnson & Navert).

In order to get rid of the influence, on the numerical solution, of layer terms out of the layer, and obtain uniform in ε convergence results it was realized that some sort of exponential fitting has to be used. These kind of methods (Il'in, Kellog & Tsan, Hemker, Dooland & Miller & Schilders) have been widely studied for one-dimensional stationary problems but their analysis on multidimensional and time dependent problems is a difficult task.

On the other hand it is well known that the use of alternating directions methods permits to generate simple (like one-dimensional) and economical schemes, for the solution of problems in mathematical phisics that involve several space variables.

In this paper we develope a new difference method which combines the well-known advantages of the one-dimensional exponential fitting methods for stationary problems and the alternating directions methods.

II. TIME SEMIDISCRETIZATION

For simplicity in the exposition of our analisys we will consider the following initial Dirichlet boundary value problem

Find $u(x,y,t)$ in $\Omega = [0,1] \times [0,1] \times [0,T]$ solution of

$$\begin{cases} \frac{\partial u}{\partial t} - \varepsilon\Delta u + \vec{v}\vec{\nabla}u + ku = f, \\ u(x,y,0) = u_0(x,y), \\ u(0,y,t) = u_1(y,t), \\ u(1,y,t) = u_2(y,t), \\ u(x,0,t) = u_3(x,t), \\ u(x,1,t) = u_4(x,t); \end{cases} \tag{1}$$

where $k(x,y,t), \varepsilon \geq 0$ and $\vec{v} = (v_1(x,y,t), v_2(x,y,t))$ such that v_i has constant sign, $i = 1,2$. We will assume compatibility between the initial and the boundary conditions, and $k, \vec{v}, u_i, f$ smooth enough, to guarantee that the solution u of (1) verifies

$$\{u, A_1u, A_2u, A_1^2u, A_2^2u, A_1A_2u, A_2A_1u\} \subset C^\infty(\Omega), \tag{2}$$

where

$$A_1 \equiv -\varepsilon\frac{\partial^2}{\partial x^2} + v_1\frac{\partial}{\partial x} + k_1,$$

$$A_2 \equiv -\varepsilon\frac{\partial^2}{\partial y^2} + v_2\frac{\partial}{\partial y} + k_2, \text{ with } k_i \geq 0 \text{ and } k_1 + k_2 = k.$$

Note that if f is smooth, conditions (2) guarantee $\{u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}\} \subset C^\infty(\Omega)$

Problem (1) is discretized in time, so that we obtain semidiscrete aproximations $u^n(x,y)$ to $u(x,y,t_n)$ solution of (1) at the instant of time $t_n = n\Delta t$ by means of the following Fractional Steps Scheme

$$\begin{cases} a)\, u^0 = u_0(x,y); \\ b) \begin{cases} (I + \Delta t A_1)u^{n+\frac{1}{2}} = u^n + \Delta t f_1(t_{n+1}), \\ u^{n+\frac{1}{2}}(0,y) = u_1(y,t_{n+1}), \\ u^{n+\frac{1}{2}}(1,y) = u_2(y,t_{n+1}); \end{cases} \\ c) \begin{cases} (I + \Delta t A_2)u^{n+1} = u^{n+\frac{1}{2}} + \Delta t f_2(t_{n+1}), (f_1 + f_2 = f), \\ u^{n+1}(x,0) = u_3(x,t_{n+1}), \\ u^{n+1}(x,1) = u_4(x,t_{n+1}). \end{cases} \end{cases} \tag{3}$$

Note that if we define "$(I + \Delta t A_1)^{-1}u$" by the resolution of the stationary problem

$$\begin{cases} (I + \Delta t A_1)z = u, \\ z(0,y) = 0, \\ z(1,y) = 0; \end{cases}$$

and analogously for "$(I + \Delta t A_2)^{-1}u$", we have that the linear operators "$I + \Delta t A_i$" are inverse monotonous and

$$\|(I + \Delta t A_i)^{-1}\|_\infty \leq 1, i = 1,2. \tag{4}$$

We define the Local Error for (3)

$$e_n = u(t_{n+1}) - \hat{u}^{n+1}, \qquad (5)$$

where $\hat{u}^{n+1}$ is the result "u^{n+1}" obtained of applying scheme (3) taking $u^n = u(t_n)$.

Under the hypothesis (2) for the solution u of the continuous problem (1), and assuming f_1 and f_2 are smooth, we obtain

$$\|e_n\|_\infty \leq C(\Delta t)^2 \text{(consistency of the scheme (3))}, \qquad (6)$$

and taking into account (6) and (4) is easy to prove

$$\|u(t_n) - u^n\|_\infty \leq C\Delta t \text{(convergence of the semidiscretization)}.$$

II. TOTAL DISCRETIZATION

Description: In order to obtain a total discrete method to approach (1), scheme (3) is discretized in space using an exponential fitting technique.

Let Ω_h be a rectangular grid (not necessarily uniform) of points $(x,y), (0, \leq x, y \leq 1)$ and let us denote $[.]_h$ the restriction of a function defined in $[0,1] \times [0,1]$ to Ω_h. Our total discretization obtains approximations u_h^n to $[u(t_n)]_h$ by means of the following algorithm

$$\begin{cases} a) \, U_h^0 = [u_0]_h; \\ b) \begin{cases} L_{1,\epsilon,h} u_h^{n+\frac{1}{2}} = u_h^n + \Delta t [f_1(t_{n+1})]_h, \\ u_h^{n+\frac{1}{2}}(0,y) = [u_1(y, t_{n+1}]_h, \\ u_h^{n+\frac{1}{2}}(1,y) = [u_2(y, t_{n+1}]_h; \end{cases} \\ c) \begin{cases} L_{2,\epsilon,h} u_h^{n+1} = u_h^{n+\frac{1}{2}} + \Delta t [f_2(t_{n+1})]_h, \\ u_h^{n+1}(x,0) = [u_3(x, t_{n+1}]_h, \\ u_h^{n+}(x,1) = [u_4(x, t_{n+1}]_h; \end{cases} \end{cases} \qquad (7)$$

where $L_{1,\epsilon,h}$ and $L_{2,\epsilon,h}$ are the result of discretizing, via exponential fitting, the uniparametric families of one-dimensional singularly perturbed eliptic problems (3)b) and (3)c) respectively

For the construction of $L_{1,\epsilon,h}$ (and similarly for $L_{2,\epsilon,h}$) on each time level t we proceed in the following way:

Let be $I_{h,y} = \{(x_0,y),(x_1,y),\ldots,(x_N,y)\} \subset \Omega_h$ with $0 = x_0 < x_1 < \ldots < x_N = 1$ the line of points in Ω_h with fix ordinate y. We note $h_j = x_j - x_{j-1}, j = 1,\ldots,N$ and $h = \max(h_j)$.

On $I_{h,y}$ we define the exponential fitting scheme in the form

$$L_{1,\epsilon,h} u_h = Q_h^{-1} T_{\epsilon,h} u_h, \text{ where}$$

$$\begin{cases} T_{\epsilon,h} u_h(x_0,y) \equiv u_h(x_0,y) = u_1(y,t), \\ T_{\epsilon,h} u_h(x_j,y) \equiv r_j^- u_h(x_{j-1},y) + r_j^c u_h(x_j,y) + r_j^+ u_h(x_{j+1},y), \\ \qquad j = 1,\ldots,N-1, \\ T_{\epsilon,h} u_h(x_N,y) \equiv u_h(x_N,y) = u_2(y,t), \text{ and} \end{cases}$$

$$Q_h g_h(x_j,y) \equiv q_j^- g_h(x_{j-1},y) + q_j^c g_h(x_j,y) + q_j^+ g_h(x_{j+1},y),$$
$$j = 1,\ldots,N-1.$$

Scheme (7) is completely determined by the coefficients r_j^-, r_j^c, $r_j^+, q_j^-, q_j^c, q_j^+$. These coefficients are obtained by imposing that the operator

$$\tau_h(w_h) = T_{\epsilon,h} w_h - Q_h w_h$$

is null on the set of functions

$$\{1, x, x^2, \exp(\frac{1}{\epsilon}\int_0^x v_1(s)ds), x\exp(\frac{1}{\epsilon}\int_0^x v_1(s)ds)\},$$

and the normalization condition

$$q_j^- + q_j^c + q_j^+ = 1.$$

In this situation, suposing h small and under the restrictions

$$h_j \leq \frac{v_1(x_j,y)\Delta t}{1 + k_1(x_j,y)\Delta t},$$

scheme (7) verifies

$$\|[\hat{u}^{n+1}]_h - \hat{u}_h^{n+1}\|_\infty \leq C\,\Delta t\, h, \qquad (8)$$

where $\hat{u}_h^{n+1}$ is the result "u_h^{n+1}" of aplying sheme (7) taking $u_h^n = [u(t_n)]_h$

Using the discrete maximum principle we have proven that, if we take homogeneous boundary conditions, we obtain for $L_{i,\epsilon,h}$, $i = 1,2$

$$\|(L_{i,\epsilon,h})^{-1}\|_\infty \leq 1, i = 1,2. \qquad (9)$$

We have also obtained results (8) and (9) with other exponential fitting methods for some other cases. for example, when there exist attractive turnig points, when the velocity field is paralel to one of the the axis or if it is null in Ω.

Finally, using the results (6),(8) and (9) it is proven

$$\|[u(t_n)]_h - u_h^n\|_\infty \leq C(\Delta t + h).$$

(uniform convergence of the totally discrete scheme)

Note that the resolution of (7)b) at each time level involves the resolution at each line of points in Ω_h of a tridiagonal sytem of linear equations (and the same for columns of points in Ω_h for (7)c)). Therefore, the computational cost is strongly decreased with respect to the standard implicit discretizations of multidimensional evolution problems

This scheme also presents other important advantages for computations. For example, in order to resolve boundary and internal layers and reduce possible numerical diffusion phenomena, this method admits mesh refinement processes that hardly increase the computational cost.

Finally we will remark that, in order to obtain a good speed-up, the algorithm can be easily implemented on paralel computers, since the resolution at each time level involves a set of uncoupled linear systems.

Numerical experiments will be presented in the oral session and the complete version of this paper.

ON NUMERICAL METHODS FOR INTERIOR SHOCK LAYER PROBLEMS

Relja Vulanović
Institute of Mathematics, University of Novi Sad
21000 Novi Sad, Yugoslavia

Abstract. A model singularly perturbed boundary value problem with a single shock layer is considered. Various numerical methods are analyzed and difficulties in resolving the layer are discussed.

The Continuous Problem

We consider the following singularly perturbed boundary value problem:

$$Tu := -\varepsilon u'' - uu' + c(x)u = 0, \quad x \in I = [0,1], \qquad (1)$$

$$Bu := (u(0), u(1)) = (U_0, U_1), \qquad (2)$$

where $0 < \varepsilon << 1$, U_0 and U_1 are given numbers, c is a sufficiently smooth function and

$$c(x) \geq c_* > 0, \quad x \in I.$$

Thus, the famous Lagerstrom-Cole model problem is included. The problem (1-2) has a unique solution u. We assume that $U_0 < 0 < U_1$, and thus $u'(x) > 0$, $x \in I$. Furthermore, let

$$u_i = \int_i^x c(t)dt + U_i, \quad i = 0,1, \qquad (3)$$

be the solutions to the reduced problem (1) with $\varepsilon = 0$, and let there exist a point $x^* \in (0,1)$ such that $u_0(x^*) + u_1(x^*) = 0$. Then u has a shock layer at $x = x^*$, cf. [1]. By using the technique from [5] we can estimate the derivatives of u:

$$|u^{(k)}(x)| \leq M[1 + \varepsilon^{-k}\exp(-m|x - x^0|/\varepsilon)], \quad x \in I. \qquad (4)$$

Here x^0 is the unique point in $(0,1)$ such that $u(x^0) = 0$, $m > 0$ is an arbitrary constant independent of ε, and by M we denote throughout a positive generic constant independent of ε. x^0 is not known usually, but it can be approximated well by x^*. Moreover, u_i, $i = 0,1$, are good approximations to u outside the layer:

$$|u(x) - u_i(x)| \leq M[\varepsilon + \exp(-m|x - x^0|/\varepsilon)], \; x \in I_i, \; i = 0,1, \qquad (5)$$

where $I_0 = [0, x^0]$ and $I_1 = [x^0, 1]$. This can be proved by using inverse monotonicity of the linear operator $Ly = -\varepsilon y'' - uy'$ with appropriate boundary operators. For $x \in I_i$ it holds that

$$LM[\varepsilon x + \exp(-\varepsilon^{-1}\int_{x^0}^x u(t)dt)] \geq \pm\varepsilon u_i(x) = \pm L[u(x) - u_i(x)],$$

and then (5) follows by the technique from [5].

Numerical Methods

The standard numerical method for solving (1-2) is discretization by the Engquist-Osher or some similar finite-difference scheme, [4]. We are interested in resolving the layer accurately, and because of (4) (with x^0 approximated by x^*) it seems reasonable to apply a special discretization mesh which is dense near x^*. This approach gives good results in the case of boundary layers, [4,5] – the pointwise errors are first order accurate uniformly in ε. Here, however, the uniform accuracy is present in the discrete L^1 norm only, and the use of the special mesh is not justified. The numerical values cluster around a point different from x^*. Another possibility is to apply an iterative mesh construction, such as the method from [2]. Neither this gives satisfactory results. The numerical solution looks fine qualitatively, but the layer is shifted from x^*. There is only one explanation for these phenomena: the discrete problem has its own layer which is shifted from the continuous one.

The best, but not ideal, results can be obtained by the technique which was applied in [3] to a boundary layer quasilinear problem. Let $k = -a\ln\varepsilon$ with a positive constant a, and let $s_0 = x^* - k\varepsilon$, $s_1 = x^* + k\varepsilon$. Then from (5) it follows that

$$|u(x) - u_i(x)| \leq M\varepsilon$$

for $x \in [0, s_0]$ if $i = 0$, and $x \in [s_1, 1]$ if $i = 1$. Thus we shall approximate u by u_0 and u_1 in corresponding intervals (u_i can be obtained from (3), either exactly or numerically), and it remains to solve the problem

$$Tv = 0, \quad x \in [s_0, s_1], \quad v(s_i) = u_i(s_i), \; i = 0,1, \qquad (6)$$

which is practically unperturbed. We shall do this by applying the standard central scheme on the equidistant mesh with the step $h = 2k\varepsilon/n$, $n \in \mathbb{N}$. The scheme is stable in the discrete L^1 norm if n is sufficiently great, independently of ε. An estimate of the error of this procedure can be obtained in L^1 norm since the operator (T, B) is L^1-stable. However, this is not sharp enough for we are interested in pointwise accuracy. The pointwise errors will be illustrated by the example with $c(x) = 1$ and $U_0 = -1$, $U_1 = 1/2$. Only the errors of the numerical solution of (6) are interesting. The numerical solution is compared to the inner solution of the problem, [1]. For $a = 5$ and $\varepsilon = 10^{-4}$ we get the errors 8.50E-2, 1.51E-2, 4.40E-3, 4.50E-3 for $n = 10, 20, 40, 80$, respectively. The errors for the same a and n but for $\varepsilon = 10^{-6}$ are 1.95E-1, 4.13E-2, 7.65E-3, 1.71E-3.

References

[1] J. Kevorkian and J. D. Cole, Perturbation Methods in Applied Mathematics, Springer, New York, 1980.

[2] B. Kreiss and H.-O. Kreiss, Numerical methods for singular perturbation problems, *SIAM J. Numer. Anal.* 18 (1981), 262-275.

[3] J. Lorenz, Combinations of initial and boundary value methods for a class of singular perturbation problems. In: *Proc. Conf. on the Numerical Analysis of Singular Perturbation Problems* (P. W. Hemker and J. J. H. Miller, eds.), Academic Press, London, 1979, pp. 295-315.

[4] R. Vulanović, Finite-difference schemes for quasilinear singular perturbation problems, *J. Comput. Appl. Math.* 26 (1989), 345-365.

[5] R. Vulanović, Continuous and numerical analysis of a boundary shock problem, *Bull. Austral. Math. Soc.* 41 (1990), 75-86.

AN ARBITRARY ORDER DIFFERENCE SCHEME FOR A NONSELF-ADJOINT SINGULAR PERTURBATION PROBLEM IN CONSERVATION FORM [1]

Pengcheng Lin
Department of Computer Science
Fuzhou University
Fuzhou, 350002, PR of China

and

Jiansheng Zheng
Xiamen International Book
Centre.
Xiamen, 361004, PR of China

Abstract A nonself-adjoint singular perturbation problem in conservation form is considered. We construct a three point difference scheme and prove that the solution of the difference scheme converges uniformly in small parameter ε with order h^{m+1} to the solution of the differential problem.

Ⅰ. INTRODUCTION

A nonself-adjoint singular perturbation problem in conservation form

$$\begin{cases} \varepsilon(p(x)u')' + q(x)u' - r(x)u = f(x), 0 < x < 1 \\ u(0) = \mu_0, u(1) = \mu_1, \end{cases} \tag{1}$$

is considered, where ε is a parameter in $(0,1]$, $q(x)$, $r(x)$, $f(x) \in w^{m+1} = \{F: F(x) \in C^m[0,1], F^{(m)}(x) \in Lipschity\}$, $p(x) \in w^{m+2}$, $0 < a_0 \leqslant \frac{q(x)}{p(x)} \leqslant b_0$, $0 < a_1 \leqslant \frac{p'(x)}{p(x)} \leqslant b_1$, $r(x) \geqslant 0$, $p'(x) \geqslant a_2 > 0$, a_0, b_0, a_1, a_2, b_1 are given constants, m is a positive integer which is given arbitrarily. Under these conditions, there is an unique solution for the equation (1).

Recently, some authors derived an accurate difference schemes of order one or twc for (1). In this paper, we construct a three-point difference scheme for (1). For any $m \geqslant 0$, we prove that the solution of the difference scheme converges uniformly in ε with order h^{m+1} to the solution of (1).

Ⅱ. THE DIFFERENCE SCHEME

A. An accurate difference scheme

We construct an accurate difference scheme for the problem (1)

$$\begin{cases} L^h u_i = v^i_{-1}(0)u_{i-1} - u_i + v^i_1(0)u_{i+1} = -v^i_0(0), \\ \qquad\qquad i = 1,2,\cdots,N-1 \\ u_0 = \mu_0, u_N = \mu_1, \end{cases} \tag{2}$$

where $v^i_k(s)$ $(k = -1,0,1)$ satisfies:

$$\begin{cases} lv^i_k(s) = p(x_i + sh)\frac{d^2 v^i_k(s)}{ds^2} + [\frac{dp(x_i + sh)}{ds} + \tau_1 q(x_i + sh)]\frac{dv^i_k(s)}{ds} \\ \qquad r(x_i + sh)\tau_2 v^i_k(s) = \tau_2 \bar{g}_k(x_i + sh), s \in (-1,1), k = \pm 1, 0, \\ v^i_1(0) = v^i_{-1}(0) = v^i_0(\pm 1) = 0, \\ v^i_1(1) = v^i_{-1}(-1) = 1, \end{cases} \tag{3}$$

here $\tau_1 = \frac{h}{\varepsilon}$, $\tau_2 = \frac{h^2}{\varepsilon}$, $g_{\pm 1}(x_i + sh) = 0$, $g_0(x_i + sh) = f(x_i + sh)$.

B. Approximate difference scheme

We establish the following approximate difference scheme.

$$\begin{cases} A^i_{-1}z_{i-1} - z_i + A^i_1 z_{i+1} = -A^i_0, i = 1,\cdots,N-1 \\ z_0 = \mu_0, z_1 = \mu_1, \end{cases} \tag{4}$$

where A^i_k $(k = -1,0,1)$ satisfies.

$$|A^i_{\pm 1} - v^i_{\pm 1}(0)| \leqslant v_{\pm 1} \leqslant \frac{\sigma h}{24}, \quad |A^i_0 - v^i_0(0)| \leqslant v_0,$$

we let

$$v^i_k(s) = \sum_{n=0}^{m} h^n v^i_{kn}(s) + h^{m+1}\varphi^i_{km}, \quad (k = -1,0,1)$$

where v^i_{kn} $(n = 0,\cdots,m)$ satisfies

$$\begin{cases} \bar{l}v^i_{kn} = p(x_i)(v^i_{kn})'' + (\varepsilon(\frac{dp(x)}{dx})|_{x=x_i} + q(x_i))\tau_1(v^i_{kn})' \\ \quad - r(x_i)\tau_2 v^i_{kn} = \tau_2 H_{kn}, \\ v^i_{k0}(-1) = v^i_{-k,0}(1) = 0, \quad k = 0,1, \\ v^i_{10}(1) = v^i_{-1,0}(-1) = 1 \\ v^i_{kn}(-1) = v^i_{kn}(1) = 0, \quad k = -1,0,1, n = 1,\cdots,m \end{cases} \tag{5}$$

here $H_{k0} = g_{k0}$ and for $n \geqslant 1$.

$$H_{kn} = \{s^n g_{kn} + \sum_{j=0}^{n-1} r_{n-j}s^{n-j}v^i_{kj} - \frac{\tau_1}{\tau_2}\sum_{j=0}^{n-1}[(n-j+1)\varepsilon p_{n-j+1} + q_{n-j}] \cdot s^{n-j}(v^i_{kj})' - \frac{1}{\tau_2}\sum_{j=0}^{n-1}s^{n-j}p_{n-j}(v^i_{kj})''\}$$

where g_{kn}, p_n, q_n, r_n are given by $y_n = \frac{y^{(n)}(x_i)}{n!}$.

C. The three-point difference scheme

We construct the three-point difference scheme for (1) such as.

$$\begin{cases} L^h u^h_i = A^{hi}_{-1}u^h_{i-1} - u^h_i + A^{hi}_1 u^h_{i+1} = -A^{hi}_0, (i = 1,\cdots,N-1) \\ u^h_0 = u_0, u^h_N = u_1, \end{cases} \tag{6}$$

where $A^{hi}_k = \sum_{n=0}^{m} h^n v^i_{kn}(0), k = -1,0,1$.

Ⅲ. THE CONVERGENCE THEOREM

We will obtain the following uniform convergence result.

Theorem Suppose u^h_i be the solution of the difference scheme (6) and $u(x_i)$ be the solution of (1), then for $0 < h < h_0$, we have

$$|u^h_i - u(x_i)| \leqslant Mh^{m+1},$$

where M is a constant independent of ε, h and i.

Ⅳ. NUMERICAL EXAMPLE

Consider the following problem

$$\begin{cases} \varepsilon(\sqrt{1+x}\,u')' + (\frac{1}{\sqrt{(1+x)}}u)' = \frac{1}{2\sqrt{1+x}}, \quad (0 < x < 1), \\ u(0) = 0, \quad u(1) = 1, \end{cases} \tag{7}$$

the exact solution is

$$u(x) = c_1\sqrt{1+x} + c_2(1+x)^{-\frac{1}{\varepsilon}} + \frac{1+x}{1+\varepsilon},$$

[1] This work supported in part by a grant of the National Science Foundation of China and in part by a grant of the Natural Science Foundation of Fujian.

where
$$c_1 = \frac{(2^{-\frac{1}{\varepsilon}} - 1 + \varepsilon)}{(1+\varepsilon)(\sqrt{2} - 2^{-\frac{1}{\varepsilon}})},$$
$$c_2 = \frac{(1 - \sqrt{2} - \varepsilon)}{(1+\varepsilon)(\sqrt{2} - 2^{-\frac{1}{\varepsilon}})},$$

$E_\infty = \max\limits_{0<i<N} |u_i^h - u(x_i)|$, here $u(x_i)$ is the solution of (7), u_i^h is the solution of the scheme (6) when m = 1. Numerical result lists [illegible] following table.

h = 0.1	j	1	6	9	E_∞
$\varepsilon = h^{\frac{1}{2}}$	$u(x_i)$ u_i^h	0.1705056 0.1705048	0.6971326 0.6971323	0.9275933 0.9275981	5.286932 E−5
$\varepsilon = h$	$u(x_i)$ u_i^h	0.2659307 0.2658965	0.7201022 0.7201026	0.9295832 0.9296028	1.7007319 E−4
$\varepsilon = h^{\frac{3}{2}}$	$u(x_i)$ u_i^h	0.3551245 0.3552783	0.7113618 0.7114487	0.9268337 0.9268539	1.856089 E−4
$h = 0.025$	j	1	26	39	E_∞
$\varepsilon = h^{\frac{1}{2}}$	$u(x_i)$ u_i^h	0.621421 E−2 0.620246 E−2	0.7535943 0.7535513	0.9825791 0.9825695	5.745888 E−5
$\varepsilon = h$	$u(x_i)$ u_i^h	0.2061851 0.2061807	0.7457681 0.7457712	0.9815736 0.9815731	4.41685 E−6
$\varepsilon = h^{\frac{3}{2}}$	$u(x_i)$ u_i^h	0.3101401 0.3101479	0.7423608 0.7423628	0.9813188 0.9813186	7.778406 E−6

The numerical results show that the numerical experiment coincides with the theoretical analysis.

The authors gratefully acknowledge the support of K. C. Wong education foundation HongKong.

References

[1] Doolan. E. P., Miller, J. J. H.. Schilders, W. H., Uniform Numerical Methods for Problems with Initial and Boundary Layers, Dublin . Boole Press, (1980).

[2] Sun Xiaedi, An arbitrary order finite difference scheme for a singular perturbation problem, Numerical Mathematics, a Journal of Chinese Universities, Vol.12, No.3 227–239 (1990).

[3] Alekceevskii, High order accurate difference scheme for a singular perturbation boundary value problem, Differential equation, Vol.17, No.7 1171–1183 (1981) (in Russian).

[4] Kellogg, R.B. and Tsan, A., Analysis of some approximation for a singular perturbation problem without turning points. Math. Comp. Vol.32, 1025–1039 (1978).

[5] K. V. Emelyanov, An accurate difference scheme for linear singular perturbation boundary problem, Report of Acadimic USSR. 1982, Vol.262, No.5, 1052–1055 (in Russian).

[6] Hegarty, A. F., J. J. H. Miller and E. O' Riordan, Uniform second order difference scheme for singular perturbation problems. Proc. Internat Conf on Boundary and Interior Layers, Computational and Asymptotic Methods (J. J. H. Miller ed) Boole Press Dublin (1980) 301–305.

[7] Guo Wen and Lin Pengcheng, An uniformly convergent second order difference scheme for a singularly perturbed self–adjoint ordinary differential equation in conservation form, Appl Math. and Mech, (English Edition) Vol.10, No.3 231–241.

THE DIFFERENCE METHOD FOR SOLVING SINGULAR PERTURBATION PROBLEMS OF THE PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS INVOLVING TWO SMALL PARAMETERS[1]

Pengcheng Lin
Department of Computer Science
Fuzhou University
Fuzhou, 350002, PR of China

and

Meifeng Yang
Fujian Branch
Cyts tours Corporation
Fuzhou, 350001, PR of China

Abstract The singular perturbation problem of the parabolic partial differential equations involving two small parameters is considered. We construct an exponentially fitted difference scheme and prove that when $t \geqslant M\mu^\delta, M\varepsilon^\delta \leqslant x \leqslant 1-M\varepsilon^\delta$ (where δ is a small positive number), the solution of the difference scheme converges uniformly to the solution of the original differential equation with order one.

I. INTRODUCTION

In this paper we consider the singular perturbation problem of the parabolic partial differential equations involing two small parameters. Wang Guoying [1] constructed a difference scheme with fitted factors. Apling the classical and non—classical estimation method to this scheme, he proved that the solution of this scheme converges uniformly to the original differential equation with $O(h^{\frac{1}{1}} + \Delta t^{\frac{1}{2}})$. In this paper, using the method of decomposing the singular term from its solution and combining an asymptotic expansion of the equation, we prove that the solution of the difference scheme converges uniformly to the solution of the differential problem with $O(h + \Delta t)$.

II. CONTINUE PROBLEM

We consider the following problem

$$\begin{cases} L_{\varepsilon\mu}u \equiv -\mu\dfrac{\partial u}{\partial t} + \varepsilon^2\dfrac{\partial^2 u}{\partial x^2} - \varepsilon a(x,t)\dfrac{\partial u}{\partial x} - b(x,t)u \\ \qquad = f(x,t),\ 0<x<1, 0<t\leqslant T \\ u_0(x) = u_0(x), \quad x\in(0,1) \\ u(0,t) = g_0(t), \quad u(1,t) = g_1(t), \quad t\in(0,T) \end{cases} \tag{1}$$

We assume:

$(H1)$: $a,b,f\in C^2(Q),\ u_0(x)\in C^4[0,1], g_0, g_1\in C^3[0,T]$

$$g_0(0) = u_0(0), \quad g_1(0) = u_0(1),$$

where

$$Q = \{(x,t), 0<x<1, \quad 0<t<T\}$$
$$\Gamma = \{x,0)z|0<x<1\}\cup\{(0,t)|0<t<T\}\cup\{(1,t)|0<t<T\}$$
$$\bar{Q} = Q\cup\Gamma.$$

(H2): $a(x,t) > \alpha > 0, b(x,t) > \beta > 0, \forall(x,t)\in[0,1]\times[0,T].$

ε, μ are positive small parameters.When $\varepsilon\to 0,\mu\to 0$ (1) is degenerated to $-b(x,t)u = f(x,t)$, therefore it lost one boundary condition in each side $x=0$, $x=1$, $t=0$. it appears boundary layer phenomenon.

Suppose problem (1) has the following form of the asymptotic solution

$$u(x,t) = \sum_{j=0}^{\infty}\sum_{i=0}^{j}\mu^{j-i}\varepsilon^i[u^0_{j-i,i}(x,t) + v_{j-i,i}(\xi_1,t) + \omega_{j-i,i}(\xi_2,t) + Z_{j-i,i}(x,\eta)$$

where $u^0_{j-i,i}$ be the solution of the degenerated problem, $v_{j-i,i}(\xi_1,t)$, $\omega_{j-i,i}(\xi_2,t), Z_{j-i,i}(x,\eta)$ are the boundary layer function near $x=0$, $x=1$ $t=0$ respectively, $\xi_1 = \dfrac{x}{\varepsilon}$, $\xi_2 = \dfrac{1-x}{\varepsilon}$, $\eta = \dfrac{t}{\mu}$.

By Vishik—Lyusternik asymptotic method we obtain (take first order approximation and let $v = v_{0,0}, \omega = \omega_{0,0}, Z = Z_{0,0}$)

$$v(\xi_1,t) = [g_0(t) - u^0(0,t) - (g_1(t) - u^0(1,t))$$
$$\cdot exp\{\frac{-[a(1,t)+\sqrt{a^2(1,t)+4b(1,t)}]}{2\varepsilon}\}]$$
$$\cdot \frac{exp\{\dfrac{[a(0,t)+\sqrt{a^2(0,t)+4b(0,t)}]x}{2\varepsilon}\}}{1-exp\{\dfrac{-[a(1,t)+\sqrt{a^2(1,t)+4b(1,t)} - a(0,t)+\sqrt{a^2(0,t)+4b(0,t)}}{2\varepsilon}\}}$$

$$\omega(\xi_2,t) = [g_1(t) - u^0(1,t) - (g_0(t) - u^0(0,t))$$
$$\cdot exp\{\frac{[a(0,t)-\sqrt{a^2(0,t)+4b(0,t)}]}{2\varepsilon}\}$$
$$\cdot \frac{exp\{\dfrac{-[a(1,t)+\sqrt{a^2(1,t)+4b(1,t)}](1-x)}{2\varepsilon}\}}{1-exp\{\dfrac{-[a(1,t)+\sqrt{a^2(1,t)+4b(1,t)} - a(0,t)+\sqrt{a^2(0,t)+4b(0,t)}]}{2\varepsilon}\}}$$

$$Z(x,\eta) = [u_0(x) - u^0(x,0)]exp\{\frac{-b(x,0)t}{\mu}\}$$

Denote

$$\lambda_{1,0} = \frac{a(0,t)-\sqrt{a^2(0,t)+4b(0,t)}}{2\varepsilon}$$

$$\lambda_{21} = \frac{a(1,t)+\sqrt{a^2(1,t)+4b(1,t)}}{2\varepsilon}$$

$$\lambda_1 = \frac{a-\sqrt{a^2+4b}}{2\varepsilon}, \quad \lambda_2 = \frac{a+\sqrt{a^2+4b}}{2\varepsilon}$$

$$a = a(x,t), \quad b = b(x,t)$$

we obtain

$$\begin{cases} |D^k_x v_{ij}(\xi_1,t)| \leqslant C\varepsilon^{-k}exp\{-\lambda_{10}x\} \\ |D^k_t v_{ij}(\xi_1,t)| \leqslant C, \quad (x,t)\in Q \\ |D^k_x \omega_{ij}(\xi_2,t)| \leqslant C\varepsilon^{-k}exp\{-\lambda_{21}x\} \\ |D^k_t \omega_{ij}(\xi_2,t)| \leqslant C, \quad (x,t)\in Q \\ |D^k_x Z_{ij}(x,\eta_2)| \leqslant C \\ |D^k_t Z_{ij}(x,\eta)| \leqslant C\mu^{-k}exp\{-\dfrac{\beta t}{\mu}\}, \quad (x,t)\in Q - v_\delta \end{cases}$$

[1] This work supported in part by a grant of the National Natural Science Foundation of China and in part by a grant of the Natural Science Foundation of Fujian

$v_\delta = \sum_{i=1}^{2} v_\delta(A_i)$, $v_\delta(A_i)$ respresent δ neighbourhood of corner (0,0) and (1,0) .

Ⅲ. DISCRETE PROBLEM

Divide region Q into rectangulaar mesh, $x_j = jh$, $j = 0,1,\cdots,J$, $h = \frac{1}{J}$, $t_n = n\Delta t, n = 0,1,\cdots,N, \Delta t = \frac{T}{N}$. Define mesh space

$$\bar{Q}^{h\Delta t} = \{(x_j,t_n)|0 \leqslant j \leqslant J,\ 0 \leqslant n \leqslant N\}$$

$$\Gamma^{h\Delta t} = \bar{Q}^{h\Delta t} \cap \Gamma, \quad Q^{h\Delta t} = \bar{Q}^{h\Delta t} - \Gamma^{h\Delta t}$$

Denote

$$D_0 g_j^n = \frac{g_{j+1}^n - g_{j-1}^n}{2h}$$

$$D_+ D_- g_j^n = \frac{g_{j+1}^n - 2g_j^n + g_{j-1}^n}{h^2}$$

$$D_t^+ g_j^n = \frac{g_j^{n+1} - g_j^{n-1}}{\Delta t}$$

where g_j^n is the function defined in $Q^{h\Delta t}$.

Take

$$\mu_j^n = \frac{\Delta t b(x_j,0) exp\{\frac{-b(x_j,0)\Delta t}{\mu}\}}{1 - exp\{\frac{-b(x_j,0)\Delta t}{\mu}\}}$$

$$\sigma_1 = \frac{-h^2 b}{4\varepsilon^2}(1 + cth\frac{h\lambda_1}{2} cth\frac{h\lambda_2}{2}),$$

$$\sigma_2 = \frac{-hb}{2\varepsilon a}(cth\frac{h\lambda_1}{2} + cth\frac{h\lambda_2}{2}),$$

where

$$\lambda_1 = \frac{a - \sqrt{a^2 + 4b}}{2\varepsilon}, \quad \lambda_2 = \frac{a + \sqrt{a^2 + 4b}}{2\varepsilon},$$

$$a = a(x_j,t_n), \quad b = b(x_j,t_n)$$

Define the difference operator

$$L^{h\Delta t}\omega^{n+1} \equiv -\mu_j^{n+1} D_t^+ \omega_j^n + \varepsilon^2 \sigma_1 D_+ D_- \omega_j^{n+1} - a(x_j,t_n)\varepsilon\sigma_2 D_0 \omega_j^{n+1} - b(x_j,t_n)\omega_j^{n+1}.$$

This operator satisfies the maximum principle

We construct the difference scheme

$$\begin{cases} L^{h\Delta t} u_j^{n+1} = f_j^{n+1} & in Q^{h\Delta t} \\ u_0^n = g_0(t_n), \quad u_J^n = g_1(t_n), & 0 < n \leqslant N - 1 \\ u_j^0 = u_0(x_j), & 1 < j < J - 1 \end{cases} \tag{2}$$

Define the mesh functions $v_j^n, \omega_j^n, Z_j^n, G_j^n$ in the following

$$\begin{cases} L^{h\Delta t} v_j^n = Lv(x_j,t_n) \\ v_j^0 = v(x_j,0) \\ v_0^n = v(0,t_n), \quad v_J^n = v(1,t_n) \end{cases}$$

$$\begin{cases} L^{h\Delta t} \omega_j^n = L\omega(x_j,t_n) \\ \omega_j^0 = \omega(x_j,0) \\ \omega_0^n = \omega(0,t_n), \quad \omega_J^n = \omega(1,t_n) \end{cases}$$

$$\begin{cases} L^{h\Delta t} Z_j^n = LZ(x_j,t_n) \\ Z_j^0 = Z(x_j,0) \\ Z_0^n = Z(0,t_n), \quad Z_J^n = Z(1,t_n) \end{cases}$$

$$\begin{cases} L^{h\Delta t} G_j^n = LG(x_j,t_n) \\ G_j^0 = G(x_j,0) \\ G_0^n = G(0,t_n), \quad G_J^n = G(1,t_n). \end{cases}$$

then $u_j^n = v_j^n + \omega_j^n + Z_j^n + G_j^n$

Ⅳ. MAIN THEOREM

Theorem 1 If $u(x,t) = v(\xi_1,t) + \omega(\mu_2,t) + Z(x,\eta) + G(x,t)$

$$u_j^n = v_j^n + \omega_j^n + Z_j^n + G_j^n$$

then when $t \geqslant M\mu^\delta$, $M\varepsilon^\delta \leqslant x \leqslant 1 \quad M\varepsilon^\delta$, $(0 < \delta < 1)$

$$\begin{cases} |G(x_j,t_n) - G_j^n| \leqslant M(h + \Delta t), \\ |v(x_j,t_n) - v_j^n| \leqslant M(h^2 + \Delta t), \\ |\omega(x_j,t_n) - \omega_j^n| \leqslant M(h^2 + \Delta t), \\ |Z(x_j,t_n) - Z_j^n| \leqslant M(h + \Delta t), \end{cases}$$

Theorem 2 Assume that the coefficient and the right hand, initial, boundary functions of (1) are sufficiently smooth,and satisfy conditions (H1) and (H2), then when $t \geqslant M\mu^\delta, M\varepsilon^\delta \leqslant x \leqslant 1 - M\varepsilon^\delta$, the solution of the difference scheme (2) converges uniformly to the soluiton of (1). as $h \to 0$, $\Delta t \to 0$, and the following estimation holds.

$$|u_j^n - u(x_j,t_n)| \leqslant M(\delta)(h + \Delta t),$$

where δ is an arbitrary number in (0,1) .

The authors gratefully acknowledge the support of K. C Wong eduction foundation HongKong.

References

[1] Wang Guoying, The difference method for solving singular perturbation problems of the parabolic partial differential equations involving several parameters, Numerical Mathematics, a journal of Chinese Universities, Vol.10, No.3, 263–272 (1988).

[2] Besjes, J. G., Singular perturbation problems for linear parabolic differential operators of arbitrary order, J. Math. Anal. Appl. 48, 594–609 (1974).

[3] Kellogg R. B. Tsan. A., Analysis of some difference approximations for a singular perturbation problem without turning points. Math. Comput. 32, 1025–1039 (1978).

SINGULAR PERTURBATION OF NONLINEAR DIFFERENCE EQUATIONS

Weijiang Zhang
Department of Applied Mathematics
Shanghai Jiao Tong University
Shanghai 200030, The People's Republic of China

Abstract: A kind of nonlinear difference equations is considered. Singular perturbation method is applied to construct the asymptotic approximation of the solution to the difference equation. Using the theory of exponential dichotomies we show that the solution of an order-reduced equation is a good approximation of the solution to the difference equation except near boundaries. The correctos, which yield asymptotic approximations, are constructed.

1, INTRODUCTION

We are considering singularly perturbed difference-boundary problems of the form

$$0=\beta(x_k)+\frac{1}{h}[f(u_{k+1})-f(u_k)]+\epsilon\frac{1}{h^2}[g(u_{k+1})-2g(u_k)+g(u_{k-1})]$$

$$k=1, \ldots, n, \quad 0<\epsilon\leq\epsilon_o \quad \text{and} \quad u_o=\alpha, \quad u_{n+1}=\beta \qquad (1)$$

They seem the "upwind" difference approximation of the following class of boundary value problems for nonlinear, second-order, ordinary differential equations

$$\epsilon g(u)_{xx}+f(u)_x+\beta(x)=0, \quad u(0)=\alpha, \; u(1)=\beta, \; x\epsilon[0, 1], \; 0<\epsilon\leq\epsilon_o$$

which has been presented in the studies of phase-locking in chains of weakly coupled oscillators as a continuum approximation ([1]).

In [2], Reinhardt proposed the numerical treatment of linear singular perturbation difference problems using formal approximations and correctors. In this paper we shall consider nonlinear difference equations. In this paper, by using singular perturbation technique, we construct two lower-order difference equations for the nonlinear difference equation (1) such that the sum of two corresponding solutions of these equations is the asymptotic approximation of that of (1). In this treatment the main difficulty is to prove that the solution of a lower-order difference equation, which is called "outer solution", is uniformly close to the solution of (1) except the boundary layer. We have proved that by using the theory of exponential dichotomies. This idea is motivated by the successful application of exponential dichotomies to multiple coupling in chains of oscillators [3]. The methods demostrated here for the nonlinear difference equation (1) can be applied to construct the combination solution for the higher-order nonlinear singular perturbation difference equations.

In [1], N. Kopell and G. B. Ermentrout have proved a proposition by which we can determine where the boundary layer of the solution to the singularly perturbed problem is. In the followings we suppose f , g and $\beta(x)$ satisfy the conditions, i.e., there is an interval J in which $g'(u)>0$, $f''(u)<0$ $f'(u)\not\approx 0$ and $Q=f(\alpha)-f(\beta)-\int_0^1\beta(s)ds>0$, for which the solution to (1) has a boundary layer on the L.H. side. For other cases, the solution has a boundary layer on the R.H. side, the similar procedure can be used to obtain the similar results.

2, REDUCED EQUATIONS AND OUTER SOLUTIONS

The reduced equations are

$$0=\beta_k+\frac{1}{h}[f(v_{k+1})-f(v_k)], \quad k=0, \ldots, n, \text{ and } v_{n+1}=\beta. \qquad (2)$$

Firstly, we denote the difference u_k-v_k by a new variable η_k and derive a discrete equation of η_k and let $\beta_k h=\omega_{k+1}-\omega_k$. Then there is such a constant Ω that

$$\Omega=\omega_k+f(v_k), \quad k=0, \ldots, n \text{ and } v_{n+1}=\beta \qquad (3)$$

This solution exists provided that v_k stay in the region in which f(u) is monotone, and hence invertible.

Similarly, we can obtain the equations for such Ω

$$\Omega=\omega_k+f(u_k)+\frac{\epsilon}{h}[g(u_k)-g(u_{k-1})] \qquad (4)$$

$k=1, \ldots, n$ and $u_o=\alpha$, $u_{n+1}=\beta$.

Using Taylor series, equations (3), (4) can be rewritten as

$$0=[f'(v_{k+1})+\frac{\epsilon}{h}g'(v_{k+1})](u_{k+1}-v_{k+1})-\frac{\epsilon}{h}g'(v_{k+1})(u_k-v_k)$$

$$+o(u_{k+1}-v_{k+1})+o(u_k-v_k)+\frac{\epsilon}{h}O(v_k-v_{k+1}) \qquad (5)$$

We note that the terms involving the form v_k-v_{k+1}, are all O (h). Thus,

$$\eta_{k+1}=A_k\eta_k+H(\eta_{k+1}, \eta_k)+O(\epsilon) \qquad (6)$$

where $A_k=\frac{\epsilon}{h}g'(v_{k+1})/[f'(v_{k+1})+\frac{\epsilon}{h}g'(v_{k+1})]$, H is at least quadratic in its variables and $k=1, \ldots, n$.

It is clear that if $f'(u)\not\approx 0$ for all $u\epsilon J$, then the absolute values of A_k are bounded uniformly away from 1. Now we wish to use this to show there are solutions u_k to (1) which stay arbitrarily close to any outer solution defined by (3) that satisfies $v_k\epsilon J$ in which f is monotone. For definiteness, we suppose $f'>0$ in J.

Theorem 1: Let { M_k } be a sequence of n×n invertible matrices, $K\epsilon Z$. Suppose that the linear difference equation $Y_{k+1}=M_kY_k$ has an exponential dichotomy on Z with constants K, σ and projections P_k.

Suppose that, for each k in Z, H is at least quadratic in its variables and $\{r_k\}$ is a bounded sequence. Then the implicit, nonlinear difference equation

$$Y_{k+1}=M_kY_k+H(Y_{k+1}, Y_k)+\epsilon r_k \qquad (7)$$

has an unique solution (Y_k) such that for sufficiently small ϵ and all $k\epsilon Z$,

$$|y_k|\leq\epsilon 2K(1+e^{-\sigma})(1-e^{-\sigma})^{-1}\sup_{k\epsilon z}|r_k|$$

Proof: The proof runs analogously to that in [4]

Theorem 1 can be used to difference equations defined for all $k\epsilon Z$ while our system (6) is defined only for $1\leq k\leq n$. We must make a suitable extension for system (6) in order to use the exponential dichotomy theory. We note that if ω_k are constants, then each outer solution v_k is also a constant, and hence the linearization A_k of (6) around v_{k+1} is also a constant. To consider (6) as an infinite system, we may define $\beta_k=0$ for $k\leq 0$ and $k\geq n+1$. Then, if ω_k are sufficiently close to a constant, then the linear homogeneous system associated with the suitable extension of (6) has an exponential dichotomy ([4]). Thus, we now establish the following theorem:

Theorem 2: Suppose that the above hypotheses. Then there is a solution to (1) that is arbitrarily close to the outer solution of (2) and

$$| u_k - v_k | = O(\epsilon) \qquad k = 1, \ldots n.$$

for sufficiently small h.

3, CORRECTORS AND ASYMPTOTIC APPROXIMATION

Suppose that in our case a boundary layer behavior occurs at $k = 0$, and there is a number h_o such that for $0 < h \leq h_o$ we have the estimate: $| u_k - v_k | \leq C\,\epsilon$, where C is independent of h and ϵ. (h_o can be determined by the roughness theorem for exponential dichotomies)

Analogously to [2], we set

$$w_k = \epsilon^k \rho_k, \qquad k = 0, \ldots, n+1 \tag{8}$$

and we can obtain

$$0 = -F\,\rho_k + \frac{1}{h} G\,\rho_{k-1}, \qquad k = 1, \ldots, n+1 \tag{9}$$

where ρ_o is a parameter which will be determined by boundary conditions, $F = f'(0)$ and $G = g'(0)$

By induction, the representation of ρ_k immediately can be obtained as follow

$$\rho_k = \frac{\rho_o}{h^k} \left(\frac{G}{F} \right)^k \qquad k = 0, \ldots, n+1$$

In order that $\{v_k + w_k\}$ presents an asymptotic approximation we choose $\rho_o = \alpha - v_o$. Finally we have

Theorem 3: suppose the above hypotheses, and $0 < m \leq | f'(u) |, | g'(u) | \leq M$ for all $u \in J$. Then $\{ v_k + w_k \}$ is an asymptotic approximation of $\{ u_k \}$ for $k = 0, \ldots, n+1$. The error satisfies the following estimates

$$| u_k - (v_k + w_k) | \leq L\,\epsilon^{1-\frac{1}{q}}$$

where L is a constant independent of ϵ and h and $q > 1$ is a constant.

Proof: Firstly, we have

$$\text{at } k = 0, \quad u_o - (v_o + w_o) = (\alpha - v_o) - \rho_o = 0$$

$$\text{at } k = n+1, \; u_{n+1} - (v_{n+1} + w_{n+1}) = \beta - (\beta + \epsilon^{n+1}\rho_{n+1}) = -\epsilon^{n+1}\rho_{n+1}$$

Generaly, we have

$$| \rho_k | \leq | \rho_o | \left(\frac{M}{hm} \right)^k, \qquad k = 0, \ldots, n+1.$$

Therefore, provided $0 < \epsilon < \epsilon_o$, where $\epsilon_o = \left(\frac{m}{M} h \right)^q < 1$, where $q > 1$, we can obtain the following estimates:

$$\begin{aligned} | u_k - (v_k + w_k) | &\leq | u_k - v_k | + | w_k | \\ &\leq C\,\epsilon + \epsilon^k | \rho_o | \left(\frac{M}{hm} \right)^k \\ &\leq \epsilon^{1-\frac{1}{q}} (C + | \rho_o |). \qquad k = 1, \ldots, n. \end{aligned}$$

since $\epsilon^k \left(\frac{M}{hm} \right)^k \leq \epsilon \frac{M}{hm} \leq \epsilon^{1-\frac{1}{q}}$ and we suppose $| u_k - v_k | \leq C\,\epsilon$, where C is a constant.

This completes the proof.

4, EXAMPLES

The first example is

$$\epsilon u'' + e^u u' - \sin\frac{x\pi}{2} = 0, \qquad u(0) = 0,\ u(1) = 0$$

It's asymptotic approximation is

$$u = Ln\left(1 - \frac{2}{\pi}\cos\frac{x\pi}{2} \right) - Ln\left(1 - \frac{2}{\pi}\exp\left(\left(\frac{2}{\pi} - 1\right)x/\epsilon\right)\right) + O(\epsilon)$$

and the solution has a boundary layer near the endpoint $x = 0$ (see Figure 1).

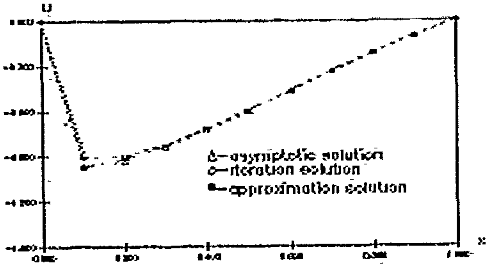


Figure 1 ($\epsilon = 0.01$ and $h = 0.02$) asymptotic solution, iteration solution and approximation solution

The second example we shall consider is

$$\epsilon u'' + e^u u' - \left(\frac{\pi}{2}\sin\frac{x\pi}{2}\right)e^{2u} = 0, \qquad u(0) = 0,\ u(1) = 0$$

This nonlinear equation is due to O'Malley [5] and the variable fitting factor is used to find the numerical solution ([6]) (see Figure 2).

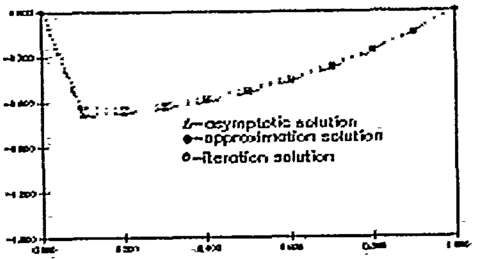


Figure 2 (ϵ=0.01, h=0.02), asymptotic solution, iteration solution and approximation solution

REFERENCES

[1] N. Kopell and G. B. Ermentrout, "Symmetry and phase-locking in chains of weakly coupled oscillators", *Comm. Pure and Appl. Math.* 39: 623-660 (1986).

[2] H. J. Reinhardt, "Singular Perturbations of Difference Methods for Linear Ordinary Differential Equations", *Applied Analysis*, 10: 53-70 (1980).

[3] N. Kopell, W. Zhang and G. B. Ermentrout, "Multiple Coupling in Chains of Oscillators", *SIAM J. Math. Anal. No.4* (1990).

[4] K. J. Palmer, "Exponential dichotomies, the shadowing lemma and transversal homoclinic points", in *Dynamical Reported*, 1: 265-306 (1988).

[5] R. E. O'Malley, Jr. , *An Introduction to Singular Perturbations*, Academic Press, New York (1974).

[6] E. P. Doolan, J. J. H. Miller and W. H. A. Schilders, *Uniform Numerical Methods for Problems with Initial and Boundary Layers*, Boole Press, Dublin (1980)

DOMAIN DECOMPOSITION TECHNIQUE FOR SINGULARLY PERTURBED PROBLEMS AND ITS PARALLEL IMPLEMENTATION

I.P. Boglaev, V.V. Sirotkin

Institute of Problems of Microelectronics Technology
USSR Academy of Sciences
Moscow District, 142432 Chernogolovka

Abstract - We are interested in numerical methods for singular perturbation problems. Iterative algorithms for domain decomposition are consided. Numerical examples are presented for both one- and two-dimensional problems. We compare the performance of a serial iterative algorithm for domain decomposition and its parallel implementation.

1. INTRODUCTION

We are interested in numerical methods for singular perturbation problems. The solution of this problem exhibits a fine structure within small regions (boundary and interior layers) of the computational domain. The traditional numerical techniques for solving singularly perturbed problems require a fine mesh covering the whole domain in order to resolve these fine local details. These methods are inefficient, since the fine mesh is not needed in those parts of the domain where the solution has a moderate variation.

We present a numerical technique where the regions of rapid change of the solution are localized in space and therefore the refinement is applied locally (near boundary and interior layers). The construction of these special meshes is based on mesh generating functions (e.g. [1]). Except the grid refinement approach, our numerical method is based on domain decomposition. The domain decomposition technique provides a natural route to parallelism.

We introduce and analyze iterative algorithms for domain decomposition which reduce the given problem to sequences of boundary value problems on each subdomain. This numerical method is illustrated by solving singularly perturbed problems for elliptic equations. We consider the case of two subdomains. However, the results given also hold for more general situations.

Firstly the problem in the one-dimensional context is discussed. The same analysis is also generalized to the two-dimensional case. Numerical examples are presented for both one- and two-dimensional problems. Here we compare the performance of a serial iterative algorithm for domain decomposition and its parallel implementation.

2. ITERATIVE ALGORITHMS

We illustrate iterative algorithms for domain decomposition for the singularly perturbed one-dimensional elliptic problem

$$L_\mu u(x) \equiv \mu^2 \frac{d^2u}{dx^2} = f(x,u), \quad x \in \Omega, \quad \Omega = (0,1), \tag{1a}$$

$$u(0)=u_0, \qquad u(1)=u_1, \tag{1b}$$

$$f_u \geq m^2 = \text{const} > 0. \tag{1c}$$

where $\mu>0$ is a small parameter. The solution of (1a)-(1c) has boundary layers at $x=0,1$ and the size of boundary layers is of the order of $h_\mu = \mu|\ln(\mu)|/m$. For simplicity, we assume that the solution $u(x)$ exhibits boundary layer only at $x=0$ (the "reduced" solution satisfies the boundary condition (1b) at $x=1$).

We introduce the overlapping decomposition of the domain Ω into two subdomains Ω_1 and Ω_2:

$$\Omega_1 = (0,\bar{x}), \quad \Omega_2 = (\underline{x},1), \quad 0<\underline{x}<\bar{x}<1. \tag{2}$$

Consider two sequences of functions $\{v^n\}$, $\{w^n\}$, $n \geq 1$, satisfying the problems:

$$L_\mu v^n(x) = f(x,v^n), \quad x \in \Omega_1, \qquad v^n(0)=u_0, \quad v^n(\bar{x}) = \bar{v}^n, \tag{3}$$

$$L_\mu w^n(x) = f(x,w^n), \quad x \in \Omega_2, \qquad w^n(\underline{x}) = \underline{w}^n, \quad w^n(1)=u_1. \tag{4}$$

We now construct two iterative algorithms.

The first one, A1, is the Schwarz alternating procedure.Here the boundary conditions $\bar{v}^n$, $\underline{w}^n$ from (3) and (4) ,respectively, are defined by

$$\bar{v}^{n+1} = w^n(\bar{x}), \ \underline{w}^n = v^n(\underline{x}), \ n \geq 1, \tag{5}$$

(the initial guess $\bar{v}^1$ should be prescribed).

The second algorithm, A2, is constructed using the interfacial problem

$$L_\mu z^n(x) = f(x,z^n), \quad x \in \Omega_{in} = (\underline{x}^*, \bar{x}^*), \qquad z^n(\underline{x}^*) = v^n(\underline{x}^*), \quad z^n(\bar{x}^*) = w^n(\bar{x}^*), \quad n \geq 1, \tag{6a}$$

where $\underline{x}^* < \underline{x} < \bar{x} < \bar{x}^*$. Here the boundary conditions from (3), (4) are determined by

$$\underline{w}^{n+1} = z^n(\underline{x}), \quad \bar{v}^{n+1} = z^n(\bar{x}), \quad n \geq 1, \tag{6b}$$

(the initial guesses $\underline{w}^1$ and $\bar{v}^1$ are given).

Algorithm A1 is a serial procedure, but algorithm A2 can be carried out by parallel processing.

For algorithms A1 and A2 we have

Proposition 1. If $\bar{x}>\underline{x}$, then iterative algorithm (3), (4), (5) converges to the solution of problem (1) with the linear rate q:

$$q=\rho(\underline{x})/\rho(\bar{x})<1, \quad \rho(x)=\mathrm{sh}(mx/\mu)/\mathrm{sh}[m(1-x)/\mu].$$

Proposition 2. Iterative algorithm (3), (4), (6) converges to the solution of problem (1) with linear rate $q<1$ provided Ω_1, Ω_2, Ω_{in} from (2), (6a) fulfill

(a) $\underline{x}-\underline{x}^*\geq 2\mu/m$, $\bar{x}^*-\bar{x}\geq 2\mu/m$;

or

(b) if μ is sufficiently small and $\underline{x}-\underline{x}^*\geq h_\mu$, $\bar{x}^*-\bar{x}\geq h_\mu$ then $q=O(\mu)$.

Remark. Iterative algorithm A2 can be generalized straightforwardly to multiple-domain decompositions.

3. NUMERICAL EXAMPLES

We present the results of some numerical experiments using the iterative algorithms A1 and A2 described in previous section.

Example 1. We consider problem (1), where $f(x,u)=1-e^{-u}$, $u_0=1$, $u_1=0$. Introduce a non-equidistant grid $\omega_x=\{x_i, 0\leq i\leq N_x\}$. The subdomains Ω_1, Ω_2 and Ω_{in} from (2), (6a) are chosen in the forms: $\underline{x}=h_\mu=x_j$, $\bar{x}=x_k$, $0<j<k<N_x$, $k-j\geq 1$, $\underline{x}^*=x_{j-1}$, $\bar{x}^*=x_{k+1}$.

In the boundary layer $[0,h_\mu]$, the mesh generating function is a logarithmic type function from [1]. We approximate the differential equation of (1a) by a simple variable-mesh difference formula. The nonlinear algebraic systems (after descretizations of (3), (4) and (6)) are solved by the one-step Newton method. In Table 1 we give the results of iterative algorithms A1 and A2 for various μ and overlapping $h=\bar{x}-\underline{x}$ values. Here the number of mesh points $N_x=101$, $j=51$ and $k\geq 52$. K_{A1} and K_{A2} denote a number of iterations for algorithms A1 and A2, respectively, to achieve an error of 10^{-5}. If we implement algorithm A2 on two parallel processors then $t_{A2}/t_0=0.525$ where t_{A2} and t_0 are execution times for algorithm A2 and for the undecomposed method from [1], respectively ($j=51$, $k=52$, $K_{A2}=K_0=4$).

TABLE 1

μ \ h	KA1			KA2		
	0.01	0.05	0.1	0.01	0.05	0.1
0.1	33	10	6	26	11	8
0.05	16	6	4	13	6	5
0.01	4	4	4	4	4	4
0.001	4	4	4	4	4	4

Example 2. We consider the two-dimensional elliptic problem

$$\mu^2\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]=1-e^{-u}, \quad (x,y)=\Omega\in(0,1)\times(0,1),$$

$$u(0,y)=\cos(\pi y/2), \quad u(1,y)=0, \quad y\in[0,1],$$

$$u(x,1)=0, \quad \left.\frac{\partial u}{\partial y}\right|_{y=0}=0, \quad x\in[0,1].$$

Introduce a non-equidistant grid $\omega=\omega_x\times\omega_y$, where ω_x as in Example 1, and ω_y is a uniform one-dimensional mesh in y. The results are presented in Table 2. Here $N_x=41$, $N_y=25$ and $j=21$, $k\geq 22$, $\Omega_{in}=(x_{j-1},x_{k+1})\times(0,1)$. If we implement algorithm A2 on two parallel processors then $t_{A2}/t_0=0.541$ ($j=21$, $k=22$, $K_{A2}=K_0=4$).

TABLE 2

μ \ h	KA1			KA2		
	0.01	0.05	0.1	0.01	0.05	0.1
0.1	59	15	9	38	14	10
0.05	31	9	6	20	8	6
0.01	7	4	4	6	4	4
0.001	4	4	4	4	4	4

A2-like algorithms can be used for solving singularly perturbed problems, where boundary and interior layers have a complex geometry. In Fig. 1 we present the solution (for $\mu=10^{-2}$) of the following problem

$$\mu^2\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]-u=\begin{cases}0, & (x^2+y^2)^{1/2}<0.5,\\ -1, & (x^2+y^2)^{1/2}\geq 0.5,\end{cases}$$

$$(x,y)=\Omega\in(0,1)\times(0,1),$$

$$u(x,0)=0, \; x\in[0,0.25], \quad u(x,1)=1, \; x\in[0,1],$$

$$\left.\frac{\partial u}{\partial y}\right|_{y=0}=0, \; x\in(0.25,1], \quad \left.\frac{\partial u}{\partial x}\right|_{x=0,1}=0, \; y\in[0,1].$$

obtained by using A2-like algorithm.

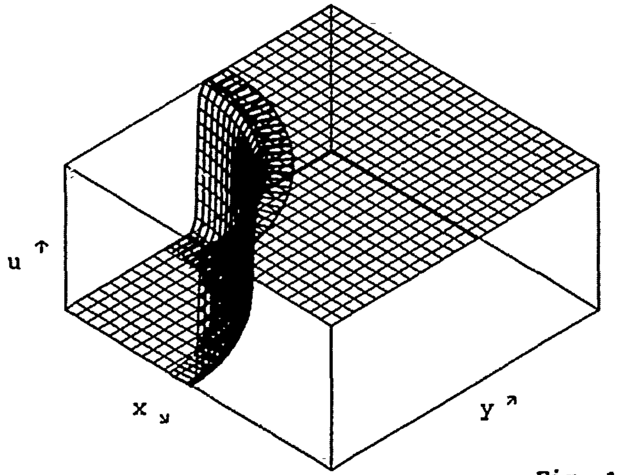


Fig. 1.

REFERENCES

[1] Boglaev,I.P.: A numerical method for a quasilinear singular perturbation problem of elliptic type . USSR, Comput. Maths. Math. Physics 28, 492-502 (1988).

SPECTRAL-FINITE ELEMENT SCHEME FOR NAVIER-STOKES EQUATIONS

GUO BEN-YU
Shanghai University of Science and Technology
Shanghai, 201800 P.R.C.

AND

CAO WEI-MING
Shanghai University of Science and Technology
Shanghai, 201800 P.R.C.

Abstract A spectral-finite element scheme is proposed for the lateral periodic and non-slip boundary problem of unsteady Navier-Stokes equations. The "inf-sup" condition is justified, and the convergence rates are presented.

I. SCHEME

Let $Q \in R^2$ be a convex polygon and $I = (0,2\pi)$. $\Omega = \{ (x,y) / x=(x_1,x_2) \in Q,\ y \in I \}$. We consider Navier-Stokes equations as follows

$$\begin{cases} \frac{\partial U}{\partial t} + (U\cdot\nabla)U + \nabla P - \nu\nabla^2 U = f, & \text{in } \Omega\times(0,T], \\ \nabla\cdot U = 0, & \text{in } \Omega\times[0,T], \\ U(x,y,0) = U_0(x,y), & \text{in } \bar\Omega, \end{cases} \qquad (1)$$

where U, P and ν are the velocity, the ratio of pressure over density and the kinetic viscosity respectively. f and U_0 are given functions with the period 2π for the variable y. We consider the lateral periodic and non-slip boundary conditions. It means that

$$\begin{cases} U(x,y,t) = 0, & \text{for } x \in \partial Q,\ y \in \bar I, \\ U(x,0,t) = U(x,2\pi,t), & \text{for } x \in \bar Q, \\ P(x,0,t) = P(x,2\pi,t), & \text{for } x \in \bar Q. \end{cases} \qquad (2)$$

In addition, the pressure satisfies the normalization condition

$$M(P) = \int_\Omega P(x,y,t)\,dxdy = 0. \qquad (3)$$

Let $C_p^\infty(\Omega)$ be the set of infinitely differentiable functions with the period 2π for the variable y. $H_p^\mu(\Omega)$ is the completion of $C_p^\infty(\Omega)$ in $H^\mu(\Omega)$ and $H_{0,p}^\mu(\Omega) = H_p^\mu(\Omega) \cap L^2(I, H_0^1(Q))$. Furthermore

$$\tilde L^2(\Omega) = \{ w \in L^2(\Omega) / M(w) = 0 \}.$$

Let $(\cdot,\cdot)$ be the scalar product in $L^2(\Omega)$ and define

$$J(u,\varphi,v) = \frac12 ((\varphi\cdot\nabla)u, v) - \frac12 ((\varphi\cdot\nabla)v, u).$$

The generalized solution of (1-3) is the pair $U(t) \in [H_{0,p}^1(\Omega)]^3$ and $P(t) \in L^2(\Omega)$ such that

$$\begin{cases} (\frac{\partial}{\partial t}U(t),v) + J(U(t),U(t),v) - (P(t),\nabla\cdot v) \\ \quad + \nu(\nabla U(t),\nabla v) = (f(t),v), \quad \forall v \in [H_{0,p}^1(\Omega)]^3, \\ (\nabla\cdot U(t), w) = 0, \quad w \in \tilde L^2(\Omega), \\ U(0) = U_0. \end{cases} \qquad (4)$$

Now, we construct the spectral-finite element scheme. For finite element approximation in the non-periodic directions, we suppose that $\{C_h\}$ is a regular family of finite triangulations of Q. Subspaces $V_h^{(q)}$ and L_h, which are composed of continuous piecewise polynomials in Q, are finite dimensional approximations to $H_0^1(Q)$ and $L^2(Q)$ respectively, q=1,2,3. On the other hand, for spectral approximation in the periodic direction, we define for any positive integer N the subspace

$$S_N = \{ \sum_{|j|\leq N} a_j e^{ijy} / a_j = \bar a_{-j} \}.$$

We choose the trial subspace for the velocity as follows

$$V_{h,N} = \{V_h^{(1)} \otimes S_N\} \times \{V_h^{(2)} \otimes S_N\} \times \{V_h^{(3)} \otimes S_N\}.$$

while the approximate pressure p is in space

$$L_{h,N} = \{L_h \otimes S_N\} \cap \tilde L^2(\Omega).$$

Let τ be the mesh size in time t and

$$u_t(t) = \frac{1}{\tau}(u(t+\tau) - u(t)).$$

The spectral-finite element scheme for solving (4) is to find the pair $u(t) \in V_{h,N}$ and $p(t) \in L_{h,N}$ such that

$$\begin{cases} (u_t(t),v) + J(u(t)+\delta\tau u_t(t),u(t),v) - (p(t) \\ \quad + \theta\tau p_t(t),\nabla\cdot v) + \nu(\nabla(u(t)+\sigma\tau u_t(t),\nabla v) \\ \quad = (f(t),v), \quad \forall v \in V_{h,N}, \\ \beta(p_t(t),w) + (\nabla\cdot(u(t)+\theta\tau u_t(t)), w) = 0, \\ \qquad w \in L_{h,N}, \\ u(0) = \Pi_{h,N}U_0, \quad p(0) = 0, \end{cases} \qquad (5)$$

where $\delta, \theta \geq 0$ and $\sigma \geq \frac12$ are parameters. The parameter $\beta > 0$ is artificial compression coefficient (see [1]). $\Pi_{h,N}$ is a projection from $[H_p^1(\Omega)]^3$ into $V_{h,N}$.

II. "INF-SUP" CONDITION AND CONVERGENCE

For convenience, we introduce firstly several non-isotropic Sobolev spaces. For $r,s \geq 0$,

$$H^{r,s}(\Omega) = L^2(I,H^r(Q)) \cap H^s(I,L^2(Q))$$

equipped with the norm

$$\|\eta\|_{H^{r,s}(\Omega)} = (\|\eta\|^2_{L^2(I,H^r(Q))} + \|\eta\|^2_{H^s(I,L^2(Q))})^{1/2}.$$

If $r,s \geq 1$, we define also

$$M^{r,s}(\Omega) = H^1(I,H^{r-1}(Q)) \cap H^{s-1}(I,H^1(Q)) \cap H^{r,s}(\Omega),$$

with the norm

$$\|\eta\|_{M^{r,s}(\Omega)} = (\|\eta\|^2_{H^{r,s}(\Omega)} + \|\eta\|^2_{H^{r-1}(I,H^1(Q))} + \|\eta\|^2_{H^{s-1}(I,H^1(Q))})^{1/2}.$$

$M_{0,p}^{r,s}(\Omega)$ is the completion of $C_p^\infty(\Omega) \cap L^2(I,H_0^1(Q))$ in $M^{r,s}(\Omega)$. Besides, we denote by $\|\cdot\|_\mu$ and $|\cdot|_\mu$ the norm and semi-norm of $H^\mu(\Omega)$, and let $\|\cdot\| = \|\cdot\|_0$.

Next, let $\nabla_x = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2})$ and $V_h = V_h^{(1)} \otimes V_h^{(2)}$. We give two assumptions for V_h and L_h as the following:

(H1): V_h and L_h satisfy the two-dimensional "inf-sup" condition, i.e., there exists a constant $\beta^* > 0$, independent of h, such that (cf.[2])

$$\sup_{\mu_h \in L_h \cap L^2(Q)} \frac{(\nabla_x \cdot v_h, \mu_h)_{L^2(Q)}}{\|\mu_h\|_{L^2(Q)}} \geqslant \beta^* |v_h|_{H^1(Q)}, \quad \forall v_h \in V_h.$$

(H2): There exist $k \geqslant 1$ and $C>0$ such that for $r \geqslant 2$ and $\bar{r} = \min(r, k+1)$,

$$\inf_{v_h \in V_h^{(q)}} |v - v_h|_{H^1(Q)} \leqslant C h^{\bar{r}-1} |v|_{H^{\bar{r}}(Q)}, \quad \forall v \in H^r(Q), \ (1 \leqslant q \leqslant 3),$$

$$\inf_{v_h \in L_h} \|v - v_h\|_{L^2(Q)} \leqslant C h^{\bar{r}-1} |v|_{H^{\bar{r}-1}(Q)}, \quad \forall v \in H^{r-1}(Q).$$

With the above two assumptions, the three-dimensional "inf-sup" condition holds for $V_{h,N}$ and $L_{h,N}$:

Lemma 1 There exists a constant $\beta > 0$, independent of h and N, such that

$$\sup_{\mu \in L_{h,N}} \frac{(\nabla \cdot v, \mu)}{\|\mu\|} \geqslant \beta |v|_1, \quad \forall v \in V_{h,N}.$$

By Lemma 1 and an error estimate of the combined spectral-finite element approximation derived from(H2), we obtain the following result.

Lemma 2 If $(U(t), P(t)) \in M_{0,p}^{r,s}(\Omega) \times H_p^{r-1,s-1}(\Omega)$ with $r \geqslant 1$ and $s \geqslant 1$, is the generalized solution of (4), $(U^*(t), P^*(t)) \in V_{h,N} \times L_{h,N}$ is its Stokes projection, i.e.,

$$\begin{cases} (\nabla(U(t) - U^*(t)), \nabla v) = 0, & \forall v \in V_{h,N}, \\ (\nabla \cdot (U(t) - U^*(t)), w) = 0, & \forall w \in L_{h,N}. \end{cases} \qquad (6)$$

then

$$|U(t) - U^*(t)|_1 + \|P(t) - P^*(t)\| \leqslant C(h^{\bar{r}-1} + N^{1-s})(\|U(t)\|_{M^{\bar{r},s}(\Omega)} + \|P(t)\|_{H^{\bar{r}-1,s-1}(\Omega)}).$$

Lemma 3 (Inverse inequality) There exists a constant $C_0 > 0$, depending only on the triangulation $\{C_h\}$, such that

$$|v|_1^2 \leqslant (C_0 h^{-2} + N^2) \|v\|^2, \quad \forall v \in V_{h,N}.$$

In order to get the convergence rates for the numerical solution of (5), we need only to estimate $|U^*(t) - u(t)|_1$ and $\|P^*(t) - p(t)\|$, where $(U^*(t), P^*(t))$ is defined in (6). By a similar analysis to that of [3], we can establish the convergence theorem as bellow.

Theorem Let (U,P) and (u,p) be solutions of (4) and (5) respectively. $U \in H^1(0,T; M_{0,p}^{r,s}(\Omega)) \cap H^2(0,T; L^2(\Omega))$, $P \in C(0,T; H_p^{r-1,s-1}(\Omega)) \cap H^1(0,T; L^2(\Omega))$, with $r \geqslant 1$ and $s \geqslant 1$. If the following conditions are fulfilled,

(i) Assumptions (H1) and (h2) hold, and $h \leqslant \frac{1}{N}$;

(ii) $\sigma > \frac{2\theta}{2\theta - 1}$ or $\delta\tau(C_0 h^{-2} + N^2) < \frac{2\theta - 1}{\sigma + \theta - 2\sigma\theta}$;

(iii) there exist $t_0 \leqslant T$ and constant $C_1 > 0$, such that

$$|U^*(0) - u(0)|_1^2 + \beta \|P^*(0) - p(0)\|^2 + C\tau \sum_{t' \leqslant t_0 - \tau} (\|U_t^*(t') - \frac{\partial U(t')}{\partial t}\|^2 + |U^*(t') - U(t')|_1^2 + \tau |U_t^*(t)|_1^2 + \beta \|P_t^*(t)\|^2) < C_1 (h^{-2} + N^2)^{-3/2} \tau^{-1} \quad ;$$

then for all $t \leqslant t_0$,

$$|U(t) - u(t)|_1^2 + \beta \|P(t) - p(t)\|^2 + \tau \sum_{t' \leqslant t - \tau} (\frac{\nu}{2} |U(t') - u(t')|_1^2) \leqslant C(\beta + \tau^2 + h^{2(\bar{r}-1)} + N^{2(1-s)}). \qquad (7)$$

Remark If $\delta = \theta > \frac{\sigma}{2\sigma - 1} > \frac{1}{2}$, then we have (7) holds for $t_0 = T$.

References

[1] Teman, R., Navier-Stokes Equations, North Holland, Amsterdam, 1977.

[2] Girault, V., Raviart, P.A., Finite Element Approximation of the Navier-Stokes Equations, Lecture Notes in Math., No.749, Springer-Verlag, Berlin, 1979.

[3] Guo Ben-yu, Scientia Sinica, 28A(1985), 1139-1153.

DETAILED NUMERICAL SIMULATION OF TWO-DIMENSIONAL IGNITION PROCESSES IN H_2-O_2 MIXTURES

U. Maas, J. Warnatz

Institut für Technische Verbrennung der Universität Stuttgart, Pfaffenwaldring 12, 7000 Stuttgart 80, Federal Republic of Germany

Abstract

New numerical methods for the solution of stiff partial differential equation systems together with the availability of fast computers with high storage capacities now allow the globally implicit simulation of instationary combustion processes in two space dimensions. Computations of ignition processes in hydrogen-oxygen mixtures are performed by solving the corresponding conservation equations (i.e., conservation of mass, energy, momentum, and species mass) using a detailed reaction mechanism (consisting of 37 elementary reactions) and a multispecies transport model. Thermal ignition is simulated by an additional source term in the energy conservation equation. Spatial discretization on a structured two-dimensional grid that is adapted statically in two spatial directions leads to large differential-algebraic equation systems which are solved numerically by an implicit extrapolation method. Results are presented for the simulation of a laser-induced thermal ignition of a hydrogen-oxygen mixture in a cylindrical reaction vessel. Due to the principal nature of the problem considered, application to many other problems seems to be possible, e.g. supersonic flow, chemical vapour deposition, atmospheric chemistry etc..

Mathematical Model

Mathematical simulation of chemically reacting multi-component compressible flow is performed by solving the corresponding system of conservation equations (Navier-Stokes equations) which may be written as [1]:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\ (\rho\, \vec{v}) = 0 \qquad (1)$$

$$\rho \frac{\partial w_i}{\partial t} + \rho\, \vec{v}\ \mathrm{grad}\ w_i + \mathrm{div}\ \vec{j}_i = \omega_i M_i \qquad (2)$$

$$\frac{\partial \rho\, \vec{v}}{\partial t} + \mathrm{grad}\ P + \mathrm{div}\ \overline{\overline{\Pi}} + \mathrm{div}\ (\rho\, \vec{v} \circ \vec{v}) = 0 \qquad (3)$$

$$\frac{\partial \rho h}{\partial t} - \frac{\partial P}{\partial t} + \mathrm{div}\ (\rho \vec{v} h) - \vec{v}\mathrm{grad}\ p + \mathrm{div}\ \vec{j}_q + \overline{\overline{\Pi}} : \mathrm{grad}\ \vec{v} = \dot{q} \qquad (4)$$

with P = pressure, T = temperature, n_s = number of species, w_i = mass fraction of species i, M_i = molar mass of species i, ω_i = molar scale rate of formation of species i, h = specific enthalpy, ρ = density, $\vec{v}$ = velocity, $\vec{j}_q$ = heat flux, $\vec{j}_i$ = diffusion flux of species i, $\overline{\overline{\Pi}}$ = stretch tensor, q = source term for deposition of energy, t = time.

The equation system is simplified by restricting the problem to two-dimensional geometries (infinite rectangular column, finite cylinder). For cylindrical geometries (considered in the example below), there are two different boundaries, namely the axis of the cylinder and the vessel surfaces. Along the axis of the cylinder symmetry boundary conditions are used. The outer boundary conditions (i.e. those at the vessel surface) are simplified by assuming non-catalytic, adiabatic walls. Nevertheless, other boundary conditions can be introduced easily in order to account for interaction of surface processes with the gas-phase reaction, as was shown for one-dimensional geometries [2].

In order to allow a detailed description of the underlying chemical and physical processes, detailed transport as well as detailed reaction models are used. Transport coefficients are computed from molecular parameters, following the kinetic theory of gases and using the *Curtiss-Hirschfelder* approximation [3,4]. Thus, the transport coefficients depend non-linearly on temperature as well as mixture composition. Thermochemical properties, namely specific enthalpies and heat capacities, both depend on temperature and are computed from polynomial fits of data from the JANAF tables [5]

The chemical reaction mechanism used for the simulation of an induced ignition of a hydrogen-oxygen mixture consists of 37 elementary reactions [2], necessary for a detailed description of the complex processes which take place during the ignition process

For the simulation of laser-induced thermal ignition in a hydrogen oxygen mixture (shown below), a source term is introduced into the energy conservation equation The energy density is assumed to decrease in axial direction and to have a Gaussian-like shape in radial direction (see [6] for details)

Transformation of the two-dimensional conservation equations into *Lagrangian* coordinates cannot be performed as easily as in the case of one-dimensional geometries, mainly due to the distortion of the grid-point system [7]. Therefore in the approach of this work, the two-dimensional instationary conservation equations are solved in *Eulerian* formulation, taking into account spatial and temporal pressure and density fluctuations.

Solution Method

The partial differential equation system describing the reacting flow consists of $n_s + 4$ partial differential equations (continuity, momentum, energy, and species conservation equations). Together with the boundary conditions it forms an initial/boundary value problem which can be solved numerically. Several properties of this partial differential equation system require special solution methods. The main problems are orders of magnitude differences in the time and length scales, and in particular the stiffness introduced by the chemical kinetics.

Spatial discretization on a rectangular mesh using finite differences leads to a system of coupled ordinary differential and algebraic equations which can be solved numerically by an semi-implicit extrapolation method [8,9] Due to the large differences of physical length scales (here particularly vessel diameter, flame front thickness, and diameter of the external energy source) adaptive gridding has to be used. In this work, the mesh is adapted statically in radial and axial direction, using a tensor product grid, i.e. a structured rectangular mesh. In the present computations, a 50 x 40 mesh is used Details of the adaptive gridding procedure can be found in [10]

Standard central difference approximations for the convective terms of the conservation equations can cause severe numerical instabilities ("over shoots") in regions of high gradients and curvatures Therefore, these terms have to be treated differently Coupling of the discretization scheme to the flow direction by use of backward and forward differencing, depending on the direction of the flow ("upwind differencing") has the disadvantage that the accuracy of the difference approximation is only of the order of the grid point distances. This leads to a large amount of numerical diffusion and thus to the flattening of steep gradients Especially in the simulation of reacting flows, such steep gradients are present in the reaction zones (e g flame fronts), and numerical diffusion would falsify the results remarkably

The approach used in this paper is based on the idea to use central difference approximations whenever possible and a modified upwind scheme, based on monotonicity preserving interpolation, otherwise. It is described in detail in [10].

Another problem arising in the numerical solution of the partial differential equation system is the resolution of shock fronts. In order to avoid severe numerical instabilities resulting if the shock is not resolved by the mesh, we apply an artificial viscosity term ("numerical diffusion") proposed by *Richtmyer and Morton* (see [7]) which spreads the shocks over a small number (typically 3) of grid points.

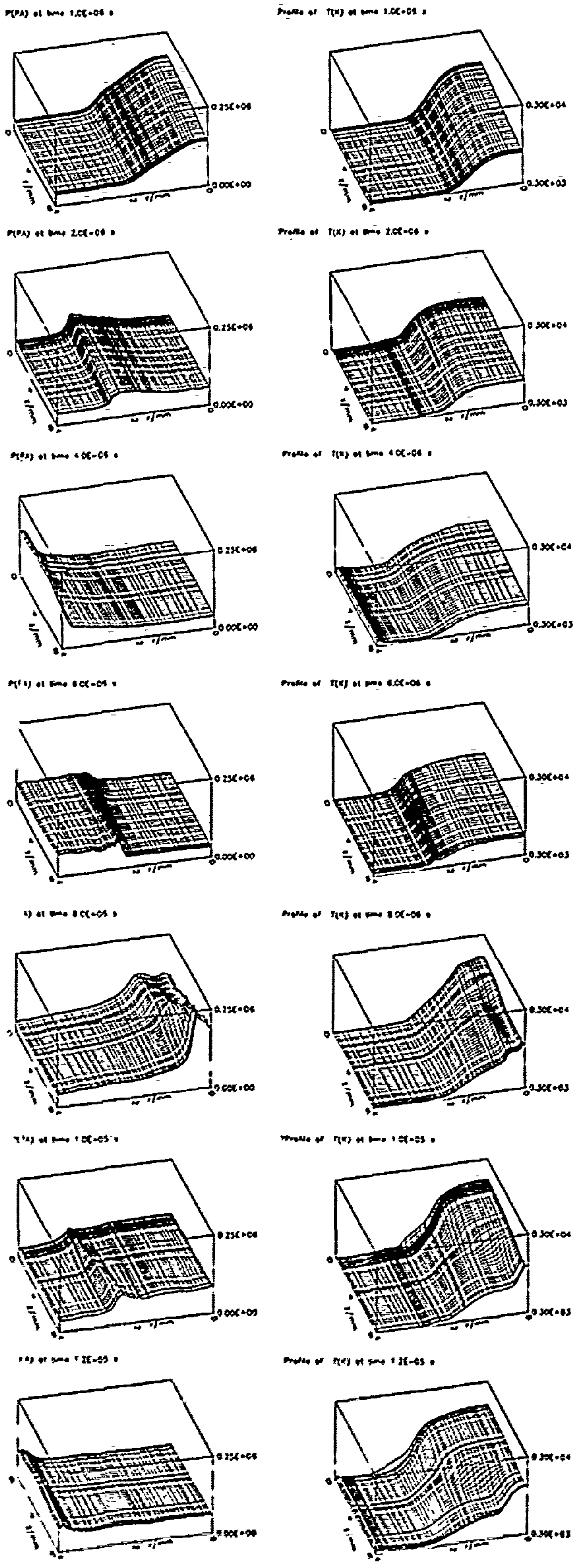
The system of ordinary differential/algebraic equations (resulting after spatial dicretization) is solved using the semi-implicit extrapolation code LIMEX [8,9]. The Jacobian matrix required for the numerical solution has a block-nonadiagonal structure. The dimension of the Jacobian is given by $n_{pde}n_ln_m$ where n_l is the number of grids in the radial, n_m the number of grids in the axial direction, and n_{pde} the number of partial differential equations. For the example shown below, the dimension of the Jacobian is 26000. The computation of the Jacobian is performed numerically by difference approximation. In order to evaluate the Jacobian in a time saving way, use is made of the block-nonadiagonal structure [10]. Due to the large dimension of the system, the solution of the linear equation systems (required by the time integration method) has to be performed by iterative methods (see [6,10] for details).

The simulation of the hydrogen-oxygen igniton (see below) takes about 50 hours on an IBM 3090; the code contains about 30,000 lines written in FORTRAN.

Results

As an example for the simulation of a chemically reacting flow, the model described above has been used to simulate a spatially two-dimensional ignition processes in a hydrogen-oxygen system with cylindrical geometry. In order to simulate induced ignition by a laser beam, thermal ignition is induced along the axis with a decreasing energy density (absorption of energy).

Spatial profiles of temperature and pressure in the reaction vessel at 1 μs, i.e. just after the external energy source has been turned off, can be seen in Figure 1. The temperature profile directly represents the spatial distribution of the ignition energy density. It decreases in axial as well as in radial direction. A similar behaviour shows the pressure, because the heating period is too short (1 μs) for the pressure to equilibrate all over the reaction volume. Thus, the pressure increase is approximately proportional to the temperature increase. The temporal development of the ignition process is shown in Figure. 1, too. At the outer boundary of the ignition volume the pressure gradient causes the formation of a shock wave moving in the radial direction and a rarefaction wave moving towards the cylinder axis. After the shock wave has reached the vessel surface (after ≈ 4 μs), it is reflected and forms a converging shock. Ignition (rapid temperature rise) occurs after a short induction period at locations where the amount of energy deposited during the heating period was high enough. Subsequently the flame front formed is moving in radial direction towards the outer boundary. Simultaneously, the flame propagates in the axial direction (in accordance with experimental results [11]). Two effects cause this flame propagation: Different induction times (depending on the local temperature) lead to a successive ignition along the axis, and at the same time a "regular" flame propagation takes place.



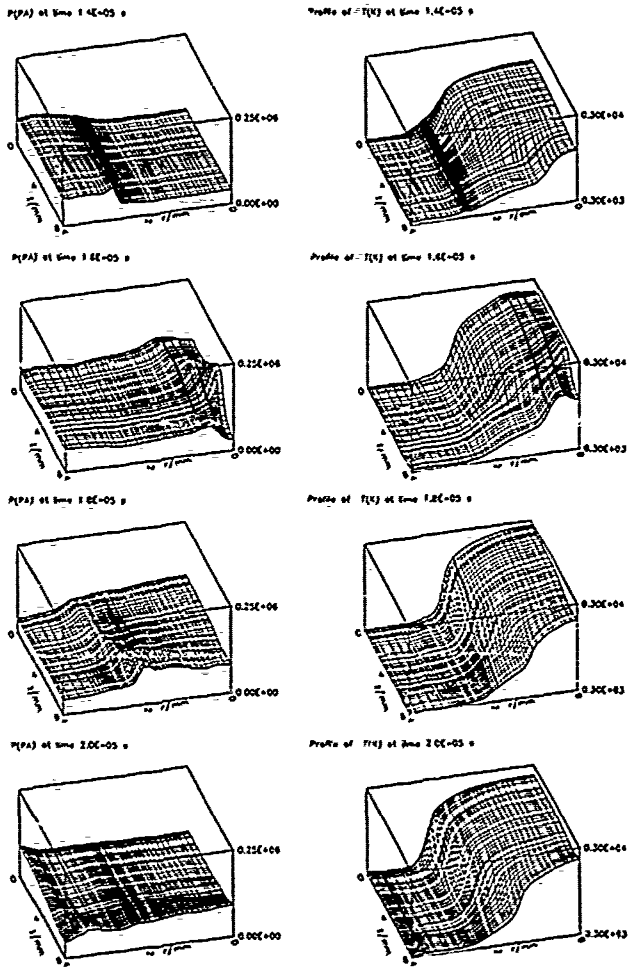


Fig. 1: Calculated time dependent profiles of pressure and temperature in an igniting stoichiometric hydrogen-oxygen mixture, cylindrical geometry, see [6] for details

Conclusions

It is possible to simulate ignition processes in hydrogen-oxygen mixtures using detailed chemistry and multi-species transport for two-dimensional geometries without simplifications like using a constant density approximation or the uniform pressure assumption. Operator splitting (which is a potential source of unreliable results) is avoided by using a fully implicit method. The method can be applied to even more complex reaction systems.

Because the hydrogen-oxygen system is an example for a chain branching ignition process, it allows an understanding of the complex interaction of chemical reaction and flow in systems of practical importance.

Furthermore, the methods described in this paper, allow the treatment of reactive flows other than those in combustion problems (e.g. supersonic flows, chemical vapour deposition etc.).

References

1. R. B. Bird, W. E. Stewart, E. N. Lightfoot, Transport Phenomena. John Wiley & Sons, Inc., New York (1960)

2. U. Maas, J. Warnatz, Combust. Flame 74, 53 (1988)

3. Hirschfelder, J.O., Curtiss, C.F. Theory of Propagation of Flames, 3rd Symp. Comb. Flame and Explosion Phenomena, 121-127, Williams and Wilkins Cp., Baltimore (1949)

4. J.O. Hirschfelder, C.F. Curtiss, R.B. Bird, Molecular Theory of Gases and Liquids, John Wiley & Sons, Inc., New York, 2nd Printing (1964)

5. JANAF Thermochemical Tables, 2nd edition. D.R. Stull, H. Prophet (Project Directors), National Bureau of Standards, Washington, D.C. (1971)

6. U.Maas, J. Warnatz. "Detailed Numerical Simulation of H_2-O_2 Ignition in Two Dimensional Geometries", Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Technical Report No. 1, University of Heidelberg (1990)

7 R. Richtmyer, K. Morton, Difference Methods for Initial Value Problems, in: L. Bers, R. Courant, J. Stoker (eds.), Interscience Tracts in Pure and Applied Mathematics No. 4, 2nd edition

8. P. Deuflhard, E. Hairer, J. Zugck, "One-Step and Extrapolation Methods for Differential/ Algebraic Systems", Num. Math. 51, 501 (1987)

9. P. Deuflhard, U. Nowak, Extrapolation Integrators for Quasilinear Implicit ODEs, in. P. Deuflhard, B. Enquist (eds.), Large Scale Scientific Computing, Progress in Scientific Computing, Vol. 7, 37. Birkhaeuser (1987)

10. U.Maas, J. Warnatz, IMPACT of Computing in Science and Engineering, 4, 354 (1989)

11 U. Maas, B. Raffel, J. Warnatz, J. Wolfrum, 21st Symposium (International) on Combustion, p. 1869. The Combustion Institute, Pittsburgh (1986)

PLANE COUETTE FLOW AS A TEST CASE FOR PHYSICO-CHEMICAL MODEL STUDIES IN HYPERSONICS

G.S.R.SARMA
DLR, Institute for Theoretical Fluid Mechanics
Bunsenstr. 10, D-3400 Göttingen, Germany

Abstract- In hypersonic flows of current and future technological interest complex reactive-diffusive processes occur in high-temperature multicomponent gas mixtures. The Couette flow lends itself as a convenient tool for testing the admittedly complicated physico-chemical models needed to understand and analyse such processes. In view of its simple geometry both the physical and mathematical aspects of the required models can be investigated through extensive and inexpensive parameter studies. Since the configuration has some features of both external flows (boundary layers on vehicles) and internal flows (in ducts and engines) such model studies yield information relevant to realistic prototype configurations. It is hoped thereby to contribute towards an understanding of the complex interaction of the various physico-chemical mechanisms involved and their relative importance so that some tractable models and simplifications for numerical and experimental studies can be identified. We illustrate our current approach through a few typical results of such studies for dissociating nitrogen and oxygen.

NOMENCLATURE

Symbols

a = molecular sound speed = $\sqrt{\gamma \mathfrak{R} T / M_2}$, c_p = specific heat at constant pressure, h = specific enthalpy, k_f, k_b = dissociation and recombination reaction rate constants, $k_{wc} = \sqrt{k_{11} \cdot k_{12}}$ at T_w, p = pressure, D = specific dissociation energy, D_{12} = binary diffusion coefficient in atom molecule mixture, K = thermal conductivity, M_1, M_2 = atomic, molecular weight, T = temperature, T_D, T_v = characteristic temperatures for dissociation and vibration; $T_D = DM_2/\mathfrak{R}$, U = relative speed between the plates, $\mathfrak{R}$ = universal gas constant, α = species concentration, δ = distance between parallel plates, η_c = temperature exponent in reaction equilibrium constant K_c, γ = adiabatic exponent = c_p/c_v, μ = dynamic viscosity; ν = kinematic viscosity; κ = thermal diffusivity = $K/\rho c_p$; ρ = density; $\theta = T_1/T_0$; τ = shear stress; τ_f = flow time scale; τ_c = chemical reaction time scale.

Subscripts

0, w: lower wall; 1: upper wall (temperature) and atomic species; 2: molecular species; ref: reference quantity.

Dimensionless parameters

Damköhler number Dam = $k_{wc}\delta(p/\mathfrak{R}T_w)^2/U = \tau_f/\tau_c$; Lewis number Le = D_{12}/κ; Mach number $Ma_w = U/a_w$; Prandtl number Pr = ν/κ; Schmidt number Sc = ν/D_{12} = Pr/Le.

I. INTRODUCTION

In view of the resurgence of interest in hypersonics for space transportation systems the basic aerothermodynamic problems of high temperature gas dynamics are under intensive study in order to assist the R & D efforts for an efficient and economic design of the proposed prototype configurations. The challenging new aspects involved in the flow fields in these configurations are related to the multicomponent reactive-diffusive gaseous mixtures present around the vehicles and in the propulsion systems [1], [2]. Since the fundamental physico-chemical processes occurring in these configurations are highly complicated and depend on a large number of uncertain and incomplete sets of experimental data and theoretical models it is felt worthwhile to investigate the effects of such inputs on some global quantities of practical interest such as heat transfer and skin friction at the vehicle walls. In this sense hypersonic Couette flow is under investigation in particular to identify the effects of transport property variations, reaction rate coefficients and boundary conditions. The pioneering work of Clarke [3] on Couette flow was based on analytically useful approximations such as constant Prandtl and Lewis numbers and Lighthill's ideal dissociating gas (IDG) model [4] and other assumptions regarding transport properties. We present here some preliminary results based on these and more general models and data for dissociating nitrogen and oxygen from [5] - [9].

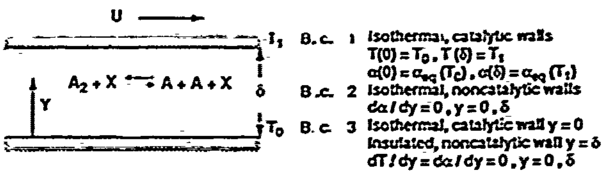

Configuration
Boundary Conditions (B. c.)
U
T_1
$A_2 + X \rightleftharpoons A + A + X$
Y
δ
T_0
B. c. 1 Isothermal, catalytic walls
$T(0) = T_0$, $T(\delta) = T_1$
$\alpha(0) = \alpha_{eq}(T_0)$, $\alpha(\delta) = \alpha_{eq}(T_1)$
B. c. 2 Isothermal, noncatalytic walls
$d\alpha/dy = 0$, $y = 0, \delta$
B. c. 3 Isothermal, catalytic wall y = 0
Insulated, noncatalytic wall y = δ
$dT/dy = d\alpha/dy = 0$, $y = 0, \delta$

Fig. 1 Dissociating diatomic gas in Couette flow.

II. PROBLEM FORMULATION AND SOLUTION

Fig. 1 illustrates the configuration and boundary conditions under consideration. We set up the general problem of a diatomic gas undergoing a dissociation/recombination reaction in a Couette flow without assuming constant Pr and Le, as was done by Clarke [3]. It consists of a system of coupled nonlinear ordinary differential equations with associated boundary conditions. The relevant boundary value problems (neglecting Soret and Dufour effects [5]) in dimensionless form are stated below. U, δ, T_w and other quantities (e. g. μ_2, K_2 etc.) evaluated at T_w are used as reference values. Mathematically, the problem retains the essential nonlinear features arising from the physico-chemical processes of interest in hypersonics. We solve the boundary value problems by a multiple shooting method in the general case using thermophysical data from [5] - [8] and by Newton-Raphson iteration in the special case of chemical equilibrium [3].

Momentum:

$$\mu \frac{\partial u}{\partial y} = \tau_w = constant \tag{1}$$

Energy:

$$K\frac{dT}{dy} + \frac{D_w}{T} R(T)\left\{ \frac{D_{12}}{(1+\alpha_1)} \frac{d\alpha_1}{dy} \right\} + E_w \mu \frac{du^2}{dy} = \mathrm{N} = constant \tag{2}$$

Species:

$$\frac{d}{dy}\left[\frac{1}{T}\left\{ \frac{D_{12}}{(1+\alpha_1)} \frac{d\alpha_1}{dy} \right\}\right] = \frac{Dam}{T^3}\left\{ \frac{2k_{11}\alpha_1 + k_{12}(1-\alpha_1)}{(1+\alpha_1)^2} \right\} \cdot \{4\alpha_1^2 - (1-\alpha_1^2)(G_w T^{\eta_c+1} \exp(-T_D/T))\}. \tag{3}$$

We may note that k_{11} and k_{12} are in general different depending on the catalytic efficiency of the atom and molecule as the respective collision partner (i. e., X = A or A_2 in Fig. 1).

Typical thermal and chemical boundary conditions are indicated in Fig. 1. The no-slip condition holds in each case: $u(0) = 0$ and $u(1) = 1$.

The dimensionless groups occurring above are defined by

$$D_w = \frac{D\rho U^2 M_2}{2K_{2w} T_w^2},\ E_w = \frac{U^2 \mu_{2w}}{2K_{2w} T_w},\ \text{and } G_w = \frac{C_e T_w^{-\eta_c+1} \mathfrak{R}}{p}.$$

G_w is related to thc *dimensional* equilibrium constant

$$K_c = C_c T^{\eta_c} \exp(-T_D/T) \tag{4}$$

by the equation

$$K_c(T) = (pG_w/\mathscr{R}T_w)T^{\eta_c} \exp(-T_D/T). \tag{5}$$

The specific enthalpy of the diatomic molecule is taken to be [3], [4], [8], [9]

$$h_2 = \frac{\mathscr{R}T}{M_2}\left[\frac{7}{2} + F_{vib}(T)\right] \tag{6}$$

where

$$F_{vib}(T) = \frac{T_v/T}{\{\exp(T_v/T) - 1\}} \tag{7}$$

is due to the harmonic oscillator vibrational energy of the molecule and contributes accordingly to molecular specific heat c_{p2} and conouctivity K_2 [3] - [5].

F(T) in eqn. (2) represents the enthalpy difference

$$(h_1 - h_2) = D \cdot F(T) = \frac{T_D\mathscr{R}}{M_2}\left[1 + \frac{3T}{2T_D} - \frac{(T_v/T_D)}{\{\exp(T_v/T) - 1\}}\right]. \tag{8}$$

IDG model [3], [4] is tantamount to assuming $F_{vib}(T) = 1/2$, $\eta_c = 0$, F(T) = 1, and G_w equal to some average value via a preassigned constant ρ_D [4].

	T_V K	T_D K	IDG	IDG-modification	
			$\rho_D\, g/cm^3$	$\rho_D(T_m)\, g/cm^3$	T_m K
Nitrogen $M_2 = 28$ $M_1 = 14$	3395	113261	130	137.4	2698
Oxygen $M_2 = 32$ $M_1 = 16$	2275	59355	150	170	1807

Table 1. Characteristic reference values used in the calculations. (Data based on values given in [4], [9]).

III. RESULTS AND DISCUSSION

A. General Model

The predominant influence for the cases considered by Clarke [3] (to be discussed later) is that due to the variation in Lewis number and the reaction rate coefficients. Variation of Pr in the temperature range considered is not signficant. On the other hand Le can vary by more than a factor of two as a function of temperature and atomic concentration. In general we do not assume Pr and Le to be constant. We solve eqns. (1) -(3) using the mixture formulas [5] -[7] for transport coefficients and reaction rate coefficient data from Wray et al. cited by Vincenti and Kruger [8], so that we have a standard dissociating diatomic gas in chemical nonequilibrium. Furthermore (assuming thermal equilibrium) we include the vibrational contribution both to the molecular enthalpy as well as specific heat. The ideal dissociating gas (IDG) of Lighthill [4] model assumes that half the vibrational states are excited and also that the equilibrium constant for the dissociation/recombination reaction is such that a characteristic constant density ρ_D related to the atomic species concentration α_{1eq} at chemical equilibrium can be defined by

$$(1 - \alpha_{1eq})/\alpha_{1eq}^2 = (\rho/\rho_D)\exp(T_D/T). \tag{9}$$

This ρ_D is a function of temperature in general and has a relative maximum at a temperature $T_m < T$, [4], [8].

Profiles of velocity, temperature and atomic concentration and variation of the heat flux H to the colder wall and the skin friction S in the general case have been discussed in [10], [11] under b. c. 1 (with U = 3.5 km/s, p = 1 atm, δ = 1 cm, T_0 = 1000 K). The profiles of T and α indicate that temperature and concentration maxima can occur in the interior (and not necessarily at the hotter wall). This is known to occur in hypersonic boundary layers [12]. In fact it is even present in Couette flow at lower speeds [13] and is due to viscous dissipation. The temperature maxima move towards the hotter wall with increasing applied temperature gradient as molecular energy transport dominates viscous dissipation. The velocity profiles deviate slightly from linearity essentially due to compressibility. A steeper rise in heat transfer rate is noted at temperatures (~ 2500 K for oxygen and ~ 4500 K for nitrogen) where appreciable dissociation begins. This is due to the availability of more energy carriers, so to say. The skin friction S on the other hand is not significantly affected by dissociation and increases mainly due to the viscosity increase with temperature (nearly linear velocity profiles contributing little to this increase).

B. Vibrational Contribution

In Figs. 2, 3 we show the results of a comparative study of models for the vibrational contribution to the heat transfer and shear stress for nitrogen (Fig. 2 (a), (b)) and oxygen (Fig. 3). In both gases vibrational modes are activated around 800 K but oxygen is fully dissociated by 5000 K when nitrogen starts dissociating appreciably [12]. The relative differences DH, DS (= absolute difference/ mean value) in heat flux H and skin friction S induced by the choice of models and approximations are illustrated in Figs. 2, 3. The reference values are those from the full equations of §II. The IDG model and three of its modifications are considered here. In all the modifications $c_{p_2} = 4\mathscr{R}/M_2$ (IDG value). In modifications two and three the equilibrium constant is that of IDG i.e.

$$K_c = \frac{4\rho_D}{M_2}\exp\left(-\frac{T_D}{T}\right),$$ each with a differen. (constant) ρ_D.

In the first modification

$$h_1 - h_2 = D \quad (\text{i.e., } T_D \gg T)$$

and K_c has the experimental (dimensional) value [8], viz.,

$$K_c = C_c T^{\eta_c} \exp(-T_D/T) \quad .$$

In the second modific'ution

$$(h_1 - h_2) = \frac{\mathscr{R}T_D}{M_2}\left(1 + \frac{T}{T_D}\right), \quad \rho_D = \rho_D(T_m)$$

i. e., vibrational contribution is averaged *without* neglecting T/T_D, and in the third modification

$$\rho_D = \rho_D(T_m), \quad h_1 - h_2 = \frac{\mathscr{R}T_D}{M_2}\left(1 + \frac{T_m}{T_D}\right).$$

The interest in the last two modifications is that the main assumption in IDG, viz., $F_{vib}(T) = 1/2$, happens to hold exactly at the temperature T_m where ρ_D has its relative maximum [14]. IDG uses a (smaller) representative value of ρ_D [3], [4]. We see that the IDG model is quite good for oxygen (Fig. 3) and the relative difference in H actually goes through zero at temperatures where the vibrational activity in oxygen is overtaken by dissociation. Here IDG overestimates the heat transfer up to ~ 2800 K and underestimates it at higher temperatures. But for nitrogen (which is vibrationally active even at higher temperatures) the IDG model is improved (cf. DH gets smaller) in modifications two and three (Fig. 2 (a)). IDG consistently underestimates H for nitrogen even for 7000 K. The deviations among the different modifications manifest themselves at temperatures where appreciable dissociation begins (Figs. 2 (a),(b)). Oxygen also shows this tendency for DH (Fig. 3).

The difference in H both for nitrogen and oxygen is a few percent in magnitude but is found to be of opposite sign. Macroscopically the representation of vibrational energy is subsumed in F_{vib} involving the ratio T_v/T_D (cf. eqns. (6) - (8)). Hence the above distinction between the two gases is to be attributed to the large difference in their respective dissociation temperatures (cf. Table 1).

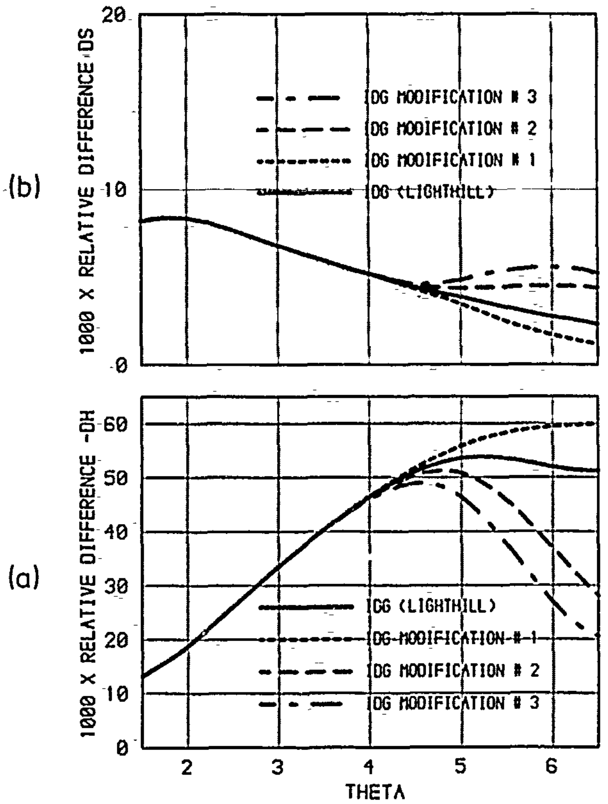


Fig. 2 Comparison of the vibrational contribution in IDG and its modifications vs. the general case under b. c. 1 : Relative difference in (a) heat flux H and (b) skin friction S for nitrogen

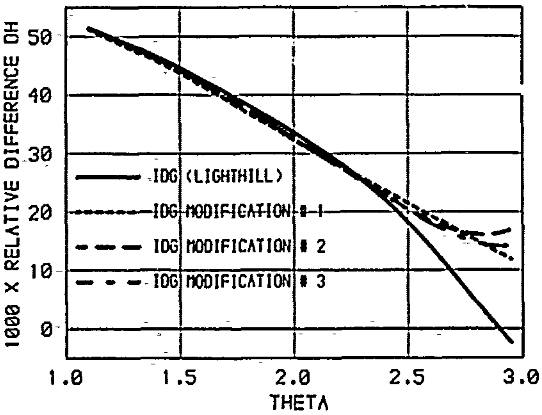


Fig. 3 Comparison of the vibrational contribution in IDG and its modifications vs. the general case under b. c. 1: Relative difference in heat flux H for for oxygen.

Relative differences in S are quantitatively even smaller than in H since skin friction is not significantly affected by dissociation or vibration but even here we find qualitative differences between the two gases. For oxygen the relative difference in S is found to steadily decrease with temperature whereas for nitrogen (Fig. 2 (b)) the different modifications show mutual deviations at temperatures where dissociation begins but vibrational modes are still active. Modification no. 1 slightly improves on IDG here.

C. Boundary conditions

Increase in heat flux H under b. c. 2 with θ is similar to that under b. c. 1 correlating with the rapid increase of dissociation for nitrogen at about 4500 K. Skin friction increases rather slowly as under b. c. 1 [11].

Under b. c. 3 the dimensional values of the global quantities H, S, and temperature T(1) at the upper insulated, noncatalytic wall and $\Lambda = \alpha_1(1)/\alpha_1(0)$ were computed [11] as functions of the lower wall temperature T_w up to 1000 K. Since the lower wall quantities are used in our nondimensionalization the computed results in dimensionless form are not easy to compare in this case. Hence the actual dimensional values were used. Heat transfer, skin friction and T_1 are found to increase with T_w almost linearly while the concentration-ratio varies very steeply in view of the negligible atomic species concentration at the low T_w values [11].

D. Transport coefficients

We now turn to the special case [3] of chemical equilibrium under b. c. 1 to illustrate the effects of tranport coefficient variation. We regard here Pr and Le as dimensionless representatives of diffusive tranport. It may also be noted that the equilibrium case serves as a useful upper bound for estimating heat transfer and was also used

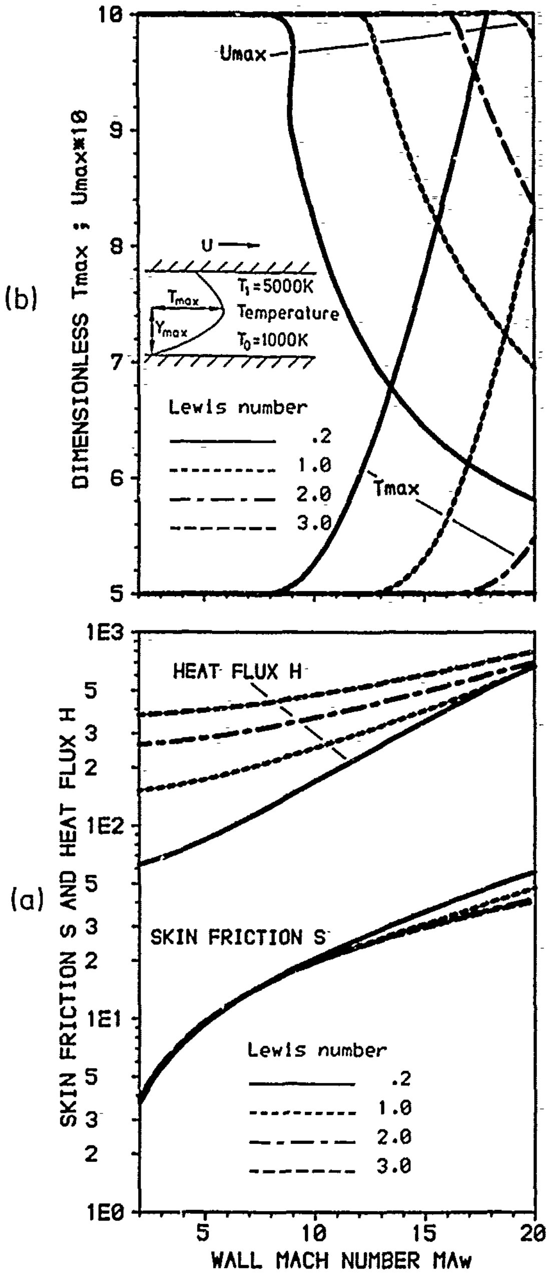


Fig. 4 Chemical equilibrium O_2 - O flow under b. c. 1. Variation of (a) heat flux H and skin friction S and (b) temperature overshoot T_{max} and corresponding velocity U_{max}, with Ma_w and Le.

as such in Space Shuttle comparisons [12]. Setting Pr = 3/4, γ = 4/3, p = 1 atm and assuming $\mu \propto \sqrt{T}$ as in [3] we compute the velocity and temperature profiles for given $\alpha_{1eq}(T)$ at equilibrium. We observe that $\alpha_{1eq}(T)$ in the flow field is compatible with the catalytic wall in b. c. 1. Clarke showed that setting constant values for Pr and Le considerably reduces the nonlinearity in the problem and allows integration of the (modified) eqn. (2). In the equilibrium case he used the IDG model for α_{1eq} whereas we test other models as well. Fig. 4 (a) shows that the heat flux in contrast to the skin friction is quite sensitive to variations of Le, especially at lower Mach numbers. Skin friction shows a slight dependence on Le at high Mach numbers. Although both H and S increase with Ma_w as to be expected their variation with Le is different. The latter difference with respect to Le may be interpreted as due to the role of diffusion (proportional to Le). Diffusion of species increases energy transport and also tends to equalize momentum, i. e., to reduce velocity gradients in the fluid layers. Thus with increasing Le skin friction is reduced. But this effect is only slight and confined to the higher Mach numbers.

As indicated in the inset of Fig. 4 (b) temperature overshoots, i.e., temperatures higher than at the hotter wall, can occur in the flow field. In Fig. 4 (b) we show the maximum temperature T_{max} and the corresponding U_{max} as functions of Ma_w and Le. Variation of U_{max} shows (since velocity profiles are almost linear) that T_{max} moves towards the middle of the flow field at all Le. Thus temperatures at which ionization and radiation would become important factors for electromagnetics and heat transfer (e. g. $T_{max} \sim 10000$ K at $Ma_w = 18$) can arise in the flow field. Under such circumstances the basic problems have to be reformulated appropriately. It is found that the temperature overshoots start to occur in nitrogen at lower Ma_w than in oxygen but are then overtaken by oxygen at higher Ma_w. This has corresponding implications for air in which nitrogen is the more abundant component.

E. Equilibrium constant

As another example of the use of Couette flow as a test case for studying physico-chemical models we show in Fig 5 the relative differences for oxygen in H (Fig. 5(a)) and T_{max} (Fig. 5(b)) as functions of Le and Ma_w with reference to the equilibrium constant in terms of ρ_D used in the IDG model [3] and that of Wray et al. cited by Vincenti and Kruger [8]. As already mentioned, the equilibrium concentration in the general case is given in terms of G_w by

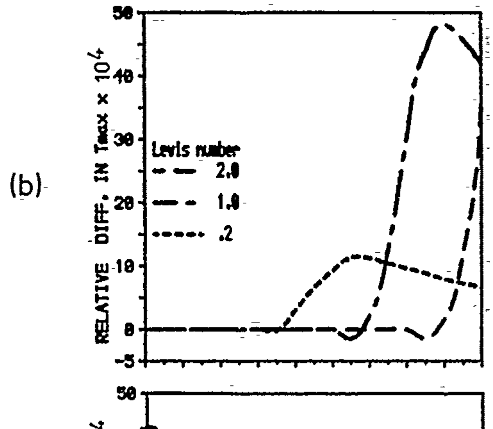


Fig. 5 Chemical equilibrium $O_2 - O$ flow under b. c. 1: Relative differences in (a) heat flux H and (b) T_{max}, between the IDG and Wray value for the equilibrium constant.

$$\alpha_1\,(equilibrium) \equiv \alpha_{1eq} = \sqrt{\frac{G_w T^{(\eta_c+1)}\exp(-T_D/T)}{4 + G_w T^{(\eta_c+1)}\exp(-T_D/T)}} \qquad (10)$$

The differences are a few percent here and are found to be even smaller for S and U_{max}. Thus the IDG model is in this sense quite useful especially for oxygen.

IV. CONCLUSIONS

IDG model offers a numerically acceptable approximation for nitrogen and oxygen for heat transfer and skin friction but the intrinsic differences between the two gases are nevertheless qualitatively discernible. Although the computations tested specific assumptions and approximations in particular examples the results shown in the simple cases considered illustrate the feasibility of assessing various physico-chemical models and role of parameters with regard to their influence on practically relevant quantities of interest in hypersonics. Such efforts should prove especially valuable in optimal use of experimental and numerical efforts through identification of trends to be expected and selection criteria for useful models. In view of the relative simplicity of the configuration geometry efficient numerical means can be devised to analyse even elaborate and complicated physico-chemical models.

REFERENCES

[1] ANON: Proc. AGARD Conference on Aerodynamics of Hypersonic Lifting Vehicles, AGARD-CP-428, 1987.

[2] C.-J. WINTER and H. SAX (Editors), "Orientierungsrahmen Hochtechnologie Raumfahrt", Report published by the German Aerospace Research Establishment, DLR, Cologne, 1987.

[3] J. F. CLARKE, "Energy transfer through a dissociated diatomic gas in Couette flow", J. Fluid Mech., vol. 4, pp. 441-465, 1958

[4] M. J. LIGHTHILL, "Dynamics of a dissociating gas. Part I. Equilibrium flow", J. Fluid Mech., vol. 2, pp. 1-32, 1957.

[5] C. R. WILKE, "A viscosity equation for gas mixtures", J. Chem. Phy., vol. 18, pp. 517-519, 1950.

[6] J. E. A. MASON, and S. C. SAXENA, "Approximate formula for the thermal conductivity of gas mixtures", Phys. Fluids, vol. 1, pp. 361-369, 1958.

[7] J. O. HIRSCHFELDER et al., Molecular Theory of Gases and Liquids, John Wiley and Sons, Inc., New York, 1964.

[8] W. G. VINCENTI and C. H. KRUGER Jr., An Introduction to Physical Gas Dynamics, John Wiley and Sons, Inc., New York, 1967.

[9] P. W. ATKINS, Physical Chemistry, 3rd Edition, Oxford University Press, Oxford, 1986.

[10] G. S. R. SARMA, "Couette Strömung eines dissoziierenden zweiatomigen Gases zwischen katalytischen Wänden", in Institutsbericht TS 1989 (Editor: W. Kordulla), IB 221-89 A 29, pp. 22-23, 1989.

[11] G. S. R. SARMA, "Couette flow of a dissociating diatomic gas", in Proc. 7. DGLR Fachsymposium, 7.-9. Nov. 1990, Aachen (to appear shortly).

[12] J. D. ANDERSON Jr., Hypersonic and High Temperature Gas Dynamics, McGraw-Hill Book Company, New York, 1989.

[13] P. A. LAGERSTROM, "Laminar Flow Theory", Chapter B in Theory of Laminar Flows (Editor: F. K. Moore), Princeton University Press, Princeton, N. J., 1964.

[14] F. K. MOORE, Chapter E: "Hypersonic Boundary Layer Theory", Chapter E in Theory of Laminar Flows (Editor: F. K. Moore), Princeton University Press, Princeton, N. J., 1964.

INTERFACE MORPHOLOGIES DURING LASER MELTING OF THIN SILICON FILMS

S. R. CORIELL and G. B. McFADDEN
National Institute of Standards and Technology
Gaithersburg, MD 20899 U.S.A.

and

L. N. BRUSH
Department of Materials Science and Engineering
University of Washington
Seattle, WA 98195 U.S.A.

ABSTRACT - Thin silicon films on a cooled substrate are often found to develop highly nonlinear interface morphologies upon radiative heating. We develop a boundary integral representation of the thermal field, and obtain numerical solutions for nonplanar solid-liquid interfaces.

I. INTRODUCTION

Radiative heating of silicon is an important processing technique for silicon wafers, especially for making silicon on insulators which are dielectrically isolated from a substrate, and also for annealing of buried layers created by ion implantation. Two-phase mixtures of solid and liquid silicon have been observed to form during the laser processing of thin silicon films on substrates such as fused silica. Furthermore, it has been observed that the morphology of the interfaces separating liquid and solid phases may either be planar or corrugated, depending upon the values of the experimental parameters. The experimental results for the two-phase silicon mixture appear to be fully consistent with the fact that liquid silicon has a higher reflectivity than solid silicon. Upon heating the silicon layer above its melting point, supercooled liquid can form adjacent to superheated solid silicon. This thermal configuration gives rise to a two-phase mixture, and leads to conditions under which planar solid-liquid interfaces within the mixture become morphologically unstable.

In this paper we extend the linear analysis by Kurtze and Jackson [*Journal of Crystal Growth* 71 (1985) 385] for a periodic array of silicon lamellae, and develop a numerical technique to treat the fully nonlinear free boundary problem.

II. NUMERICAL METHODS AND RESULTS

Jackson and Kurtze consider a two dimensional model for heat flow in the film consisting of two dimensional diffusion equations with source terms that result from the imposed heat fluxes J through the plane surfaces of the film; in each phase the source term is taken to be a constant. The diffusion equations thus assume the form

$$\frac{1}{\alpha_L}\frac{\partial T_L}{\partial t} = \frac{\partial^2 T_L}{\partial x^2} + \frac{\partial^2 T_L}{\partial y^2} - J_L, \qquad (1)$$

for the liquid phase, and, for the solid phase,

$$\frac{1}{\alpha_S}\frac{\partial T_S}{\partial t} = \frac{\partial^2 T_S}{\partial x^2} + \frac{\partial^2 T_S}{\partial y^2} + J_S; \qquad (2)$$

here the quantities α_L and α_S are the thermal diffusivities of the liquid and solid, respectively.

At a crystal-melt interface the Gibbs-Thomson condition,

$$T_L = T_S = T_M - T_M \Gamma K, \qquad (3)$$

gives the equilibrium melting temperature at a curved interface, where $\Gamma = \gamma / L$ is a capillary coefficient, γ is the surface tension, L is the latent heat of fusion per unit volume of solid phase, and K is the mean curvature of the interface. Conservation of heat at the interface requires

$$k_L \nabla T_L \cdot \hat{n} - k_S \nabla T_S \cdot \hat{n} = -L v_n, \qquad (4)$$

where $\hat{n}$ is the unit normal vector at the interface pointing into the liquid phase. k_L and k_S are thermal conductivities, and v_n is the normal velocity of the interface.

Steady solutions to the nonlinear governing equations may be expressed in terms of boundary integrals, which also provide an accurate and efficient computational procedure. This effectively reduces the dimensionality of the problem from two to one, and allows the calculation of interface shapes that are not easily expressed as a single-valued function. Implementation of the boundary integral technique for the treatment of arbitrarily shaped interfaces is facilitated by using a relative arclength representation for the solid-liquid interface. We set $e = s/S_T$, where s is the arclength along the interface and S_T is the total arclength of the interface over a full period. We then describe the interface parametrically as the set of points $\{x(e), y(e)\}$ for $0 < e < 1$; the functions $y(e)$ and $[x(e) - e]$ are both periodic functions of e. Given an interface shape, the temperature at a point $z' = x' + iy'$ in the liquid can be written in the form

$$T_L(z') = \tfrac{1}{2} J_L (y - \mathcal{H}_L)^2 + \int_\gamma \sigma_L(z) \frac{\partial G_L(z,z')}{\partial n} ds, \qquad (5)$$

where

$$G_L(z,z') = \frac{1}{2\pi} \log|\sin\pi(z - z')| + \frac{1}{2\pi} \log|\sin\pi(z - z'')| \qquad (6)$$

is a periodic Green's function with reflection symmetry about the line $y = \mathcal{H}_L$; here the image point z'' is given by $z'' = x' + i(2\mathcal{H}_L - y')$. The normal derivative and the integration in Eqn. (5) are performed with respect to the unprimed variables, and s is the arclength along the interface curve γ over a single period. The outward normal to the curve γ is given by the vector $(y_e, -x_e)/S_T$. The unknown dipole distribution σ_L is determined by the Dirichlet boundary conditions (3) for T_L. An integral equation of the second kind for σ_L is obtained by letting z' tend to a point on the boundary γ; this equation is discretized and inverted to give an approximation to the dipole distribution σ_L. Having found σ_L, we compute the normal derivative $\partial T_L / \partial n$ required in the conservation of heat condition at the interface, Eqn. (4). A similar procedure is used to compute $\partial T_S / \partial n$. In general, for arbitrary interface shapes, the heat flux equation is not satisfied, so iteration of Eqn. (4) using Newton's method is applied to find the unknown interface $\{x(e), y(e)\}$. For each updated guess of the interface shape, the entire integral equation solution method is repeated in order to find new values of $\partial T_L / \partial n$ and $\partial T_S / \partial n$. Provided the initial guess is sufficiently close, the procedure quickly converges to an interface shape consistent with all equations and boundary conditions.

We choose to parametrize the interface by the tangent angle $\phi(e)$,

$$\phi(e) = \tan^{-1}\left(\frac{dy}{de} / \frac{dx}{de}\right). \qquad (7)$$

and express $\phi(e)$ as a Fourier series,

$$\phi(e) = \sum_{n=0}^{\infty} (a_n \cos 2n\pi e + b_n \sin 2n\pi e). \qquad (8)$$

Given the function $\phi(e)$, the interface shape $(x(e), y(e))$ can be computed from the $\phi(e)$ and the initial values $x(0) = 0$ and $y(0)$ by quadrature.

For numerical purposes we retain a finite set of Fourier coefficients for $\phi(e)$ by truncating the infinite series at $n = M$. The boundary integral equations are then discretized using the trapezoid rule, this results in a numerical approximation with spectral accuracy. The heat conservation boundary condition (4) is also discretized to give a system of nonlinear equations in the unknowns a_n, b_n, and $y(0)$.

We have used values appropriate to silicon in our numerical calculations. Periodic solutions having wavelength λ are computed most efficiently by choosing the computational domain to have length $\lambda/2$ in the x-direction, with no-flux boundary conditions applied at $x = 0$ and $x = \lambda/2$, we achieve this in practise by imposing this symmetry on the Fourier coefficients, setting $a_n = 0$. On this domain one may compute solutions which are periodic with wavelength λ, and also solutions which are higher harmonics with wavenumbers which are integer multiples of $\omega = 2\pi/\lambda$.

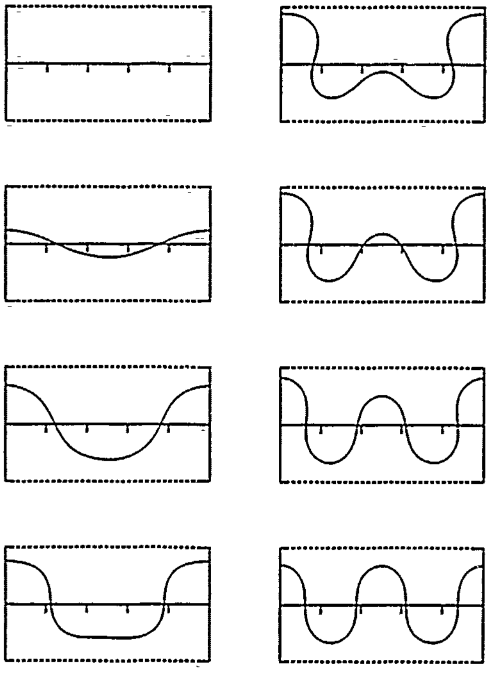


Figure 1. A sequence of interface shapes tracked along the λ solution branch.

Fig. 1 shows a sequence of interface shapes calculated using a continuation method starting with a planar shape at the onset of instability of a perturbation with wavelength equal to λ. We define the amplitude A of an interface shape to be the square root of the sum of the squares of the Fourier coefficients for the tangent angle, so that the amplitude is zero for the planar state. As the solution branch is traced out to larger amplitudes, nonlinear effects cause higher order harmonics (in particular the first harmonic) to develop. The fundamental component gives way to the appearance of its first harmonic, and ultimately disappears entirely.

The computational domain also allows solutions with wavenumbers that are integer multiples $n\omega$ of the fundamental wavenumber ω; these solutions bifurcate from the base state at progressively higher values of the power J_L. The result of tracking several such solution branches is shown in Fig. 2, which is a plot of the amplitude of the steady state interfacial shapes as a function of the power J_L. We also calculate numerically the linear stability of each of the computed nonlinear steady states. On the plot, stable nonlinear interface shapes are represented by solid curves while the dashed curves represent unstable interface shapes. The planar state first loses stability to the fundamental mode ($n = 1$) at $J_L = 1.7(10^7)$ K/cm^2, which bifurcates supercritically. This fundamental solution branch increases in amplitude with increasing power until it reaches a limit point near $J_L = 3.4(10^7)$ K/cm^2, after which it increases in amplitude with decreasing power; this branch loses stability at the limit point. It terminates at finite amplitude at a secondary bifurcation point on the nonlinear solution branch corresponding to solutions with period $\lambda/2$ ($n = 2$). This $\lambda/2$ family of solutions bifurcates subcritically from the planar solution at $J_L = 3.5(10^7)$ K/cm^2, and the family is initially unstable to two types of perturbations, for small amplitudes, the unstable disturbances are approximately sinusoidal with wavenumbers ω and 2ω. The $\lambda/2$ family reaches a limit point near $J_L = 2.3(10^7)$ K/cm^2, where it sheds one mode of instability, and then encounters the secondary bifurcation point with the fundamental solution branch, where the remaining mode of instability is lost, the $\lambda/2$ family is then stable for increasing values of the power. The $\lambda/3$ family bifurcates subcritically at $J_L = 7.0(10^7)$ K/cm^2, and initially has three modes of instability. The $\lambda/3$ family also has a limit point at $J_L = 3.0(10^7)$ K/cm^2, which is followed by two secondary bifurcation points, after which it continues to larger power as a stable solution branch. To avoid further complicating the figure, only the primary solution branches are shown, and the solution branches issuing from these secondary bifurcation points are omitted.

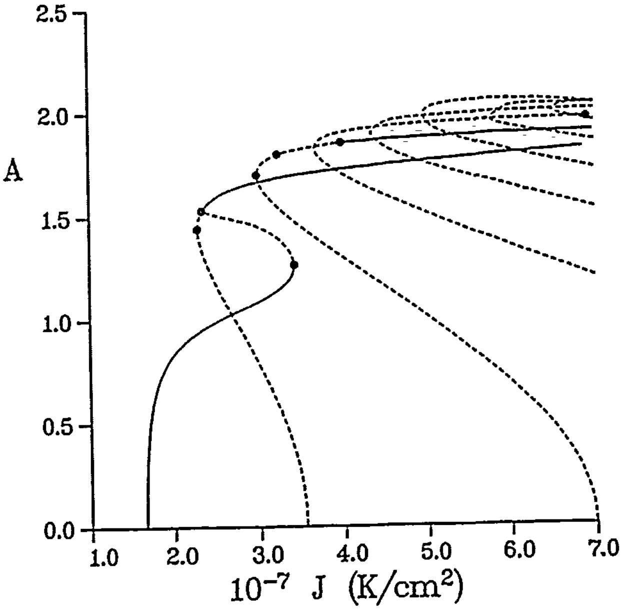


Figure 2. Bifurcation diagram of interface amplitude A versus the power J_L for solutions branches corresponding to integer multiples $n\omega$ of the fundamental wavenumber ω.

Further primary solution branches for the families with periods $\lambda/4$ through $\lambda/8$ are also shown in this figure; their points of bifurcation occur off-scale. Although the separation between these bifurcation points from the planar state are increasing well-separated, they are also progressively more subcritical, and have limit points that all occur within the range shown in the figure with amplitudes at the limit point in the range $1.5 < A < 2$. In each case the family with period λ/n initially has n modes of instability, and the trend of shedding instabilities at secondary bifurcation points once the limit point is passed also seems to hold; to avoid further complicating the figure the secondary bifurcation points are not shown for the primary modes with $n > 3$. The primary mode with $n = 4$, for example, regains stability at $J_L = 6.9(10^7)$ K/cm^2, as indicated by a dot in the figure.

We are indebted to K. A. Jackson for suggesting this problem. This work was conducted with the support of the Microgravity Science and Applications Division of NASA, and the Applied and Computational Mathematics Program of DARPA.

ACOUSTIC PROPAGATION IN THE OCEAN-SURFACE BUBBLE LAYER[1]

MICHAEL J. BUCKINGHAM

Marine Physical Laboratory
Scripps Institution of Oceanography
La Jolla, CA 92093, U.S.A.

and

The Institution of Sound and Vibration Research
The University
Southampton, SO9 5NH, England

Abstract Wave-breaking events on the surface of the ocean entrain air, which creates a layer of bubbles immediately beneath the surface. The sound speed in the layer increases monotonically with depth, forming a surface duct in which sound may propagate in the form of normal modes. Since sound is currently being used to study a number of surface processes, including gas transfer across the air-sea interface, the effect of the profile on the observations is of some concern. An analytical theory of sound propagation in surface ducts has recently been developed, based on an inverse-square profile, which provides an exact solution for the acoustic field in terms of a sum of normal modes. On the basis of the theory, it is possible to obtain the solution of certain inverse problems; for example, once the parameters of the profile have been prescribed, the source depth can be estimated from the width of the modulation structure that appears in the acoustic spectrum at the receiver. The theory also provides an interpretation of the acoustic signatures of wave-breaking events observed in the La Perouse and FASINEX experiments.

I. INTRODUCTION

When an ocean wave breaks, air is entrained and forced below the sea surface, where it fragments into a large number of micro-bubbles. Acoustically, the bubbles have two important effects. for a few milliseconds after the instant of closure a bubble rings, that is to say, it oscillates in the radial or breathing mode, hence acting as a very effective acoustic source, this being the mechanism which is responsible for much of the sound produced by breaking waves, and the presence of the large population of bubbles reduces the speed of sound immediately below the surface by as much as 20 m/s. The sound-speed profile down through the bubble layer is monotonic increasing, which gives rise to upward refraction, thus creating the condition necessary for the formation of a surface waveguide. The sounds from breaking waves are trapped in this bubbly waveguide and hence undergo dispersion, which manifests itself in the form of well-defined features in the spectrum observed at a receiver in the duct [1].

An analytical model of sound propagation in the ocean-surface bubble layer has been developed [2], based on the so-called inverse-square profile:

$$\frac{1}{c^2(z)} = \frac{1}{c_\infty^2}\left(1 + \frac{d^2}{z^2}\right), \quad z \geq z_s \ , \qquad (1)$$

where c(z) is the speed of sound as a function of the depth coordinate, z, the parameter d provides a measure of the effective depth of the duct, and c_∞ is the asymptotic value of the sound speed in the limit of infinite depth. In the coordinate scheme of Eq. (1), the origin of z lies above the sea surface, which falls at $z = z_S$. Below the surface, the ocean is treated as a semi-infinite medium, and the surface itself is assumed to be a plane, pressure-release boundary. The most significant features of the new theory are described briefly below, with emphasis placed on their relevance to the interpretation of wave-breaking spectra.

II. PROPAGATION IN AN INVERSE-SQUARE DUCT

The Helmholtz equation can be solved exactly for the harmonic field from a point source in an inverse-square channel bounded above by a pressure-release surface and extending below to infinity. The solution takes the form of a branch line integral, representing short-range radiation, and an infinite sum of normal modes. Only the modal component of the field is significant in the present context.

Immediately below the sea surface each mode shows oscillatory behaviour, down as far as the *extinction depth*. The oscillatory region is where the modal energy is concentrated. Below the extinction depth the oscillations cease and the mode exhibits an exponential decay to zero. The oscillatory and evanescent regions are illustrated in Fig. 1,

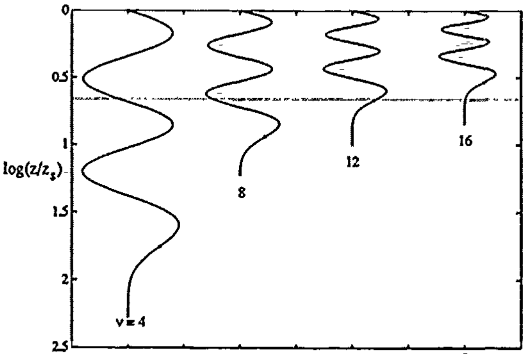


Figure 1. The fifth mode at progressively increasing frequencies, as indicated by v which scales with f. See the text for further details.

which shows the fifth mode at four different frequencies. The asterisk on each curve depicts the extinction depth, which becomes shallower as the frequency rises. Thus, at a fixed depth, as indicated by the horizontal dotted line in Fig. 1, the phenomenon of *mode drop-out* occurs. below the *drop-out frequency* the extinction depth falls below the line, so a receiver on the line detects the mode, but as the frequency increases and the extinction depth passes upwards through the line, the same receiver delivers a null response since the mode is now in extinction. The drop-out frequency of the mode is defined as that frequency for which the extinction depth is coincident with the receiver on the line. Obviously, the drop-out frequency depends on the depth of the line and on the mode number.

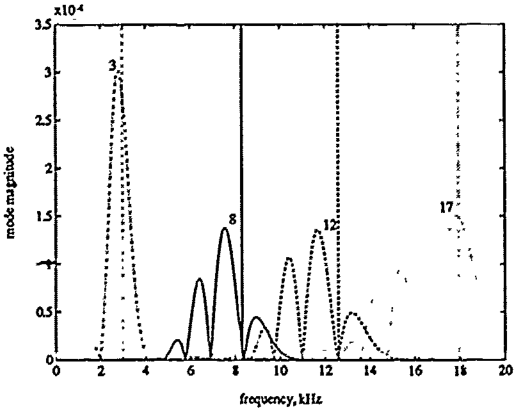


Figure 2. Spectral shape of modes 3, 8, 12 and 17 at a fixed depth. The vertical lines indicate the drop-out frequencies of the respective modes.

Fig.2 shows the magnitude of several modes at a fixed depth as a function of frequency. In this example, the parameters of the inverse-square profile are representative of the surface bubble layer in the FASINEX experiment [1]. Each mode occupies a finite bandwidth, within which it displays several well-defined peaks. It is evident that on performing a coherent modal synthesis, to obtain the power spectrum of the field, interference between the overlapping regions of the modes will occur. Thus, the resultant spectrum will inevitably show a complicated structure of peaks and troughs. It is reasonable to suppose that a measured spectrum of sound (from breaking waves or any other source) in a surface duct should be similarly complex.

In the analysis of such spectra, the mode drop-out frequency would seem to be a useful interpretive tool, and indeed it is, as Farmer and Vagle [1] have demonstrated; but, at best, it is rather crude. This observation is illustrated in Fig. 2, where it can be seen that the drop-out frequency, depicted by the vertical lines, falls close to the principal

[1] Research supported by the United States Office of Naval Research, under contract N00014-91-J-1118.

maximum in the odd-order modes but is almost coincident with the highest zero in the even-order modes. In general, the drop-out frequency is not an accurate indicator of the position of the principal maximum in the spectrum of a mode, differing perhaps by 10% or so, as in modes 8 and 12 in Fig. 2. When a number of modes are superposed, this discrepancy is compounded by the fact that each mode shows secondary maxima which may be comparable in magnitude with the principal maximum, making interpretation of the peaks in the spectrum doubly difficult. Both problems are eliminated by computing the full theoretical spectrum for comparison with the experimental observations.

III. WAVE-BREAKING SPECTRA

Farmer and Vagle [1] recorded the sound of breaking waves at two locations, identified as La Perouse and FASINEX, with a shallow hydrophone at a depth of 14 m and 24 m, respectively. A surface bubble layer was present in both cases, its depth, as measured by the parameter d in Eq. (1), being a factor of 2.5 greater in FASINEX than La Perouse. The observations were made over the frequency band from 1 to 20 kHz.

The observed spectra from the two locations are different in structure. that from La Perouse shows half a dozen well-defined, relatively narrow peaks, whereas the FASINEX spectra show several broad bands of energy separated by distinguishable gaps. These features are believed to be genuine, rather than artifacts of the instrumentation, because they are repeatable over a number of wave-breaking events but with minor variations due to the slowly changing nature of the bubbly surface duct (D. M. Farmer, private communication, September 1990).

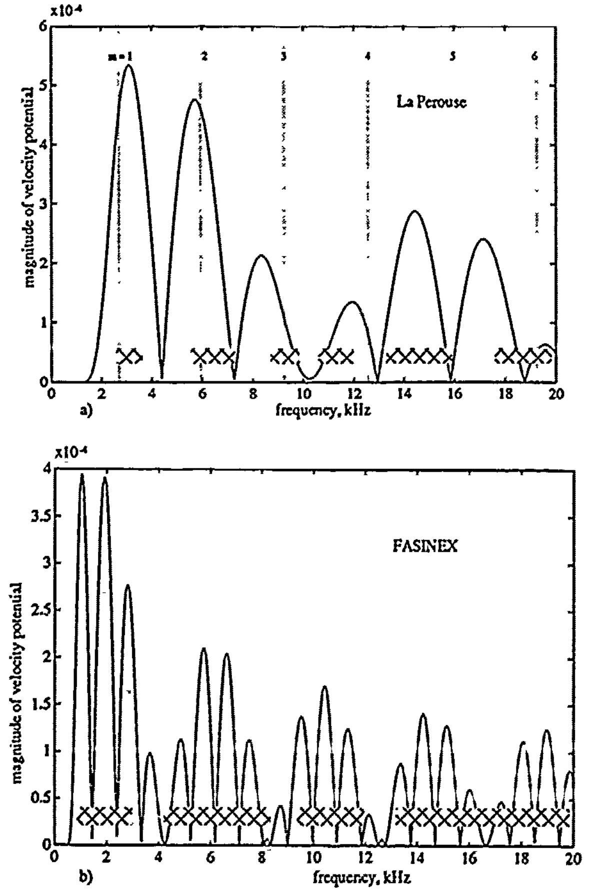


Figure 3. Inverse-square spectra (solid lines) for a) La Perouse and b) FASINEX. The cross-hatching indicates observed spectral peaks in a) and spectral bands in b). The vertical dotted lines in a) depict the drop-out frequencies of the indicated modes.

Fig. 3a shows the spectrum (solid line) of a wavebreaking event at La Perouse, as computed from the inverse-square theory using channel parameters obtained from measurements [1] of the sound speed profile through the bubble layer. For comparison, the cross-hatched, horizontal stripes indicate the spectral maxima observed by Farmer and Vagle [1]. Notice the isolated spectral maxima and the fidelity of the match between theory and experiment.

Similarly, Fig. 3b shows the inverse-square theory (solid line) compared with experimental observations (cross hatching) for FASINEX, the deeper of the two surface channels. Again, the theoretical spectrum contains well-defined maxima, but in this case clustered into groups, or bands, separated by regions of little or no energy. The positions of the bands closely match the experimental observations [1], as indicated by the cross hatching.

The detailed agreement between the inverse-square theory and experiment suggests strongly that in La Perouse and FASINEX the observed spectral structure is an effect of propagation through the upward-refracting surface bubble layer. An alternative proposal [3], that the spectral features are an effect of the source (non-linear bubble oscillations), may be valid in other situations, where the surface bubble layer is not well developed, but in the La Perouse and FASINEX data sets the propagation does appear to dominate the spectrum.

IV. SPECTRAL PEAKS AND SPECTRAL BANDS

The two spectra in Fig. 3 show well-defined maxima that are obviously related to the spectral peaks in the individual modes. In Fig. 3b, these peaks are strongly modulated, to form bands which are separated by gaps, where little energy exists. In fact, a similar modulation is present in Fig. 3a, but so much slower that it is barely noticeable. (The spectral period of the modulation is inversely proportional to d, the effective depth of the channel, which is larger in FASINEX than in La Perouse by a factor of 2.5).

Apart from d, the modulation also depends on the source depth, z'. If Δf₀' is the spectral period of the modulation (*i.e.* the distance between the bands), then the following approximate relationship holds:

$$\Delta f_0' \approx \frac{c_\infty}{2d\,\ln\left(\frac{z'}{z_*}\right)} . \qquad (2)$$

It follows that, once the parameters of the profile are known, it is possible to perform an inverse calculation to determine the depth, z', of the source from a measurement of $\Delta f_0'$.

On applying this argument to the FASINEX data, a source depth of 1.5 m is obtained, and this is the value that was used in the computation of Figs. 3a and 3b. Now, this is unexpectedly deep for acoustically active bubbles. The bubble layer is known to extend down to several metres, but most of the constituent bubbles are mature and hence quiescent. It is, perhaps, possible that the roughness of the sea surface and the non-uniformity of the bubble layer could be factors in leading to an exaggerated estimate of the depth of the bubble sources. Although this cannot be discounted, the fact that such good agreement between theory and experiment is observed in Fig. 3 suggests that the bubbly duct does act as a deterministic waveguide with a planar- pressure-release boundary, at least over the short ranges considered here. Such behaviour would not be expected if non-uniformity in the channel, for whatever reason, were a significant factor.

V. CONCLUDING REMARKS

The inverse-square theory of sound propagation through an upward-refracting surface channel shows excellent agreement with observations [1] of wave-breaking spectra obtained at two locations with a near-surface hydrophone. This agreement supports the conclusion that the spectral structure present in the observed wave-breaking signatures is introduced as the signal propagates through the bubbly surface duct.

REFERENCES

1. D. M. Farmer and S. Vagle, "Waveguide propagation of ambient sound in the ocean-surface bubble layer", J. Acoust. Soc. Am., 86, 1897-1908 (1989).
2. M. J. Buckingham, "On acoustic transmission in ocean-surface waveguides", to be published in Phil. Trans. Roy. Soc., June 1991.
3. M. S. Longuet-Higgins, "Bubble noise spectra", J. Acoust. Soc. Am. 87, 652-661 (1990).

MARCHING TECHNIQUES BASED ON ELLIPTIC WAVE EQUATIONS

George H. Knightly* and Donald F. St.Mary†
Center for Applied Mathematics and Mathematical Computation
Department of Mathematics
University of Massachusetts
Amherst, MA 01003, USA

Abstract-A class of methods for marching acoustic waves forward in range is presented, based on a far field elliptic approximation to the Helmholtz equation. The stability of the methods is analyzed and two particular schemes are examined in more detail. Explicit stability criteria are obtained for these two schemes, amounting to easily verified restrictions on the step sizes. The stability criteria are affirmed in sample computations.

I. INTRODUCTION

We are concerned here with the problem of determining the acoustic wave due to a source of given frequency, ω, in the ocean waveguide. We suppose the solution is known (e.g. by measurement) out to some range, r_0, and we wish to propagate the solution forward to larger ranges. To simplify the discussion we consider here only the azimuth-independent case and utilize the range coordinate r and depth coordinate z. Thus, we investigate the following problem for the Helmholtz equation with variable wave speed $c(r,z)$, pressure release boundary at the surface $z=0$ and hard bottom $z=B$.

$$\nabla^2 p + \kappa^2(r,z)p = 0, \;\; r_0 < r, \; 0 < z < B, \qquad (1a)$$

$$p(r,0) = 0, \;\; \frac{\partial p}{\partial z}(r,B) = 0, \qquad r_0 < r, \qquad (1b)$$

$$p(r,z) \text{ given}, \qquad r < r_0, \qquad (1c)$$

where $\kappa(r,z) = \omega/c(r,z)$.

When backscatter is unimportant, parabolic equation methods are widely used (e.g., see [5] and references cited therein) to march the solution forward in range. Here we continue some investigations [1, 3, 4, 6] into marching with a far field elliptic equation that retains the potential to include backscatter.

To develop a far field elliptic approximation to problem (1), let k_0 denote a reference wave number and $H_0^{(1)}$ the usual Hankel function satisfying

$$H_0^{(1)}(k_0 r) \approx \sqrt{\frac{2}{\pi k_0 r}} e^{i(k_0 r - \pi/4)}, \quad k_0 r \gg 1. \qquad (2)$$

If we now make the substitution $p = w(r,z)H_0^{(1)}(k_0 r)$, and use the far field property (2), then (1) leads to the problem

$$w_{rr} + 2ik_0 w_r + w_{zz} + k_0^2 \phi w = 0, \qquad (3a)$$

$$w(r,0) = \frac{\partial w}{\partial z}(r,B) = 0, \; r_0 < r, \qquad (3b)$$

$$w(r,z) \text{ given}, \; r \le r_0, \; 0 \le z \le B, \qquad (3c)$$

where

$$\phi(r,z) = [\kappa(r,z)/k_0]^2 - 1. \qquad (4)$$

Problem (3) is ill-posed! There are solutions arbitrarily small at $r = r_0$ but arbitrarily large at $r = r_0 + \delta$ for small $\delta > 0$. Yet, as seen in [1, 3, 4, 6], marching methods for problem (1) can be successful. In the next section we outline some marching schemes for (1), in section III we derive stability criteria for the schemes and examine two particular schemes in more detail. Some results of computations testing the criteria are presented in section IV.

II. MARCHING SCHEMES

In this section we develop a family of marching schemes for problem (3) by (i) introducing Pade rational approximations for an operator in (3a), (ii) discretizing the resultant approximate problem and (iii) specifying a technique for solving the discrete problem.

We first rewrite the far field elliptic equation (3a) as

$$(\frac{\partial}{\partial r} + ik_0)^2 w - (ik_0)^2[\phi + 1 + \frac{1}{k_0}\frac{\partial^2}{\partial z^2}]w = 0 \qquad (5)$$

and use a Padé rational approximation $[\frac{P(h)}{Q(h)}]^2$ for the operator

$$1 + h = (\sqrt{1+h})^2, \qquad (6)$$

where

$$h = \phi + \frac{1}{k_0^2}\frac{\partial^2}{\partial z^2}.$$

We apply the chosen approximation in (5) and operate with Q^2 to obtain

$$Q^2 w_{rr} + 2ik_0 Q^2 w_r + k_0^2(P^2 - Q^2)w = 0. \qquad (7)$$

We specify mesh points by choosing a range step size dr and a depth step size $dz = B/M$, for a chosen integer M. Set

$$w_n^m = w(r_0 + ndr, mdz), \;\; m = 0,1,\ldots,M+1; \; n = -1,0,1,2,\ldots$$

and

$$w_n = (w_n^1, \ldots, w_n^M)^T, \;\; n = -1,0,1,2,\ldots$$

The boundary conditions are discretized as $w_n^0 = 0$ and $w_n^{M+1} = w_n^{M-1}$.

Various marching schemes can be obtained by making different choices of the operators P, Q and their discretizations P_n, Q_n and from using different methods to solve the discrete system for w_{n+1}. For example, if we use central differences to approximate w_r and w_{rr}, then (7) leads to a discrete system

$$\begin{aligned} 0 =& Q_n^2(\frac{w_{n+1} - 2w_n + w_{n-1}}{(\Delta r)^2}) \\ &+ 2ik_0 Q_n^2(\frac{w_{n+1} - w_{n-1}}{2\Delta r}) + k_0^2(P_n^2 - Q_n^2)w_n. \end{aligned} \qquad (8)$$

where P_n and Q_n are $M \times M$ matrices.

We illustrate the above approach by specifying two such schemes.

Scheme I is the basic marching method, close to that developed in [3], obtained by making the choices

$$P^2(h) = 1 + h, \;\; Q^2(h) = 1. \qquad (9)$$

Then equation (8) becomes

*This research supported in part by ONR Contract N00014-90-J-1031.

†This research supported in part by ONR Contract N00014-91-J-1119.

$$0 = \frac{w_{n+1} - 2w_n + w_{n-1}}{(\Delta r)^2} + 2ik_0(\frac{w_{n+1} - w_{n-1}}{2\Delta r}) + k_0^2 h_n w_n, \quad (10)$$

where h_n is the $M \times M$ tridiagonal matrix having m^{th} diagonal entry $\phi_n^m - (2/(k_0\Delta z)^2)$ and all entries in the adjacent diagonals are $a_0 = (k_0\Delta z)^{-2}$, except the entry in row M column $M-1$ is $2a_0$. Equation (10) is easily solved explicitly for w_{n+1} in terms of w_n and w_{n-1}. Since w is given for $r \leq r_0$, w_{-1} and w_0 are known and enable the marching to proceed.

Scheme II is obtained using

$$P(h) = 1 + \frac{3}{4}h, \quad Q(h) = 1 + \frac{1}{4}h, \quad (11)$$

so that $\frac{P(h)}{Q(h)}$ is the usual Padé(1,1) approximation to $\sqrt{1+h}$. Then P_n^2 and Q_n^2 are 5-diagonal matrices. We use a 5-diagonal system solver in this case to solve (8) for w_{n+1}.

III. STABILITY

Each of the schemes, I and II, leads to an interpretation as

$$W_{n+1} = \mathcal{M}_n W_n. \quad (12)$$

Here $W_n = (w_n, w_{n-1})^T$ is a $2M$-vector and $\mathcal{M}_n$ is a $2M \times 2M$-matrix of the form

$$\mathcal{M}_n = \begin{bmatrix} \eta T_n & -\tau I \\ I & O \end{bmatrix} \quad (13)$$

with

$$\eta = \frac{1}{1 + ik_0\Delta r}, \quad \tau = \frac{1 - ik_0\Delta r}{1 + ik_0\Delta r} \quad (14)$$

$$T_n = (2 + k_0^2(\Delta r)^2)I - k_0^2(\Delta r)^2(Q_n^2)^{-1}P_n^2 \quad (15)$$

For stability, we require that all cigenvalues, λ, of $\mathcal{M}_n$ satisfy

$$|\lambda| \leq 1 \quad (16)$$

The eigenvalues of $\mathcal{M}_n$ are obtained from the eigenvalues, x, of T_n by the relation

$$0 = \lambda^2 - \eta x \lambda + \tau. \quad (17)$$

For schemes I and II the x's are real and one derives from (14), (16) and (17) the stability condition

$$x^2 < 4(1 + (k_0\Delta r)^2). \quad (18)$$

For both methods I and II in the constant coefficient case the x's can be found explicitly and the stability condition (18) yields the following restrictions on the step sizes. These limits gave reliable guidelines for choices of step sizes in the test problems.

$$\text{Scheme I:}\Delta z > \frac{2}{k_0}, \quad \Delta r < \Delta z[(\frac{\Delta z k_0}{2})^2 - 1]^{\frac{1}{2}} \quad (19)$$

$$\text{Scheme II:}\Delta z > \frac{\sqrt{3}}{k_0}, \quad \Delta r < \frac{1}{2k_0}[(k_0\Delta z)^2 - 3] \quad (20)$$

IV. CALCULATIONS

We illustrate the implications and validity of the mesh restrictions (19),(20) through some test calculations performed with schemes I and II. The test problem is Part B of "test Case 2 (Bilinear Profile)" taken from the *NORDA Parabolic Equation Workshop* ([2], p.36). For this problem $k_0 = 0.9$ and the most conservative estimates from (19),(20) give

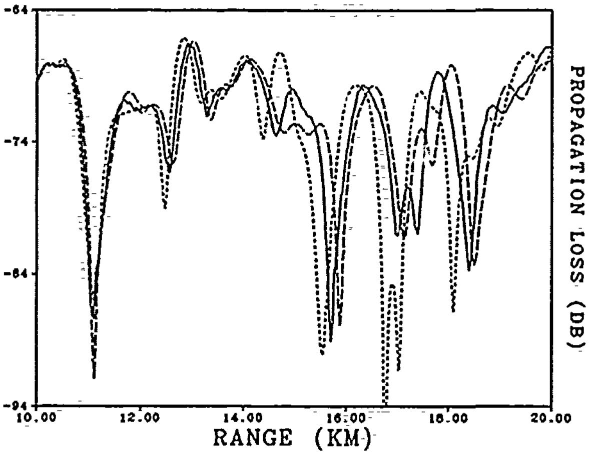


FIGURE 1.

$$\text{Scheme I:} \quad \Delta z > 22.2m \quad (21)$$

$$\text{Scheme II:} \quad \Delta z > 19.2m. \quad (22)$$

With $\Delta z = 25m$ in Scheme I, $\Delta r < 12.9m$ is required by (19). With $\Delta z = 20m$ in Scheme II, $\Delta r < 9.95m$ is required by (20). Figure 1 shows propagation-loss curves for this problem, based on calculations using Scheme I (dashed curve) with $\Delta z = 25m$, $\Delta r = 1m$ and using Scheme II (solid curve) with $\Delta z = 20m$, $\Delta r = 3m$. The reference solution (dotted curve) was obtained using the higher order methods of [5]. Note the improvement of Scheme II over Scheme I (at the cost of some small high frequency oscillation). The Scheme I curve did not improve significantly as Δz was decreased below $25m$. In fact, the Scheme I calculation became unstable for Δz in the vicinity of the limiting value $22.2m$ given in (21).

References

[1] K. J. Baumeister, Numerical spatial marching techniques in duct acoustics, *J. Acoust. Soc. Am.* 66, 297-306 (1979).

[2] J A. Davis, D. White, and R. C. Cavanagh, NORDA Parabolic Equation Workshop, 31 March - 3 April 1981, *NORDA Technical Note 143*, NSTL Station, MS, 1982.

[3] G. H. Knightly, and D. F St. Mary. Marching methods for elliptic models of underwater sound propagation. *Computational Acoustics: Wave Propagation*, D. Lee.,et al., eds., North-Holland, New York, 397-407.

[4] G. H. Knightly, and D. F. St. Mary. Computational Ocean Acoustics, to appear *Proceedings of the Second Edward Bouchet International Conference on Physics and Technology*, Accra, Ghana, 1990.

[5] G. H. Knightly, D. Lee, and D. F. St. Mary. A higher order parabolic wave equation. *J. Acoust. Soc. Am.* 82, 580-587 (1987).

[6] V. Y. Zavadskii, Y. S. Kryukov, Finite-difference calculation of sound fields in an irregular ocean sound channel, *Sov. Phys. Acoust.*, 29, 449-453 (1983).

TREATMENT OF HORIZONTAL DENSITY VARIATIONS IN A 3-DIMENSIONAL OCEAN

DING LEE AND GEORGE BOTSEAS
Naval Underwater Systems Center
New London, CT 06320
U.S.A.

WILLIAM L. SIEGMANN
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
U.S.A.

Abstract - A technique for handling vertical density variations in a three-dimensional ocean has been introduced by Lee-Schultz-Saad. It is generally believed that the acoustic effect of density variation in the vertical direction is stronger than that of a density variation in the horizontal direction. We extended the above treatment to handle horizontal density variations in a 3-dimensional ocean by the same numerical technique and incorporated this extended technique into the FOR3D (a 3-dimensional wave propagation prediction code). This updated code was applied to examine the effect between vertical and horizontal density variations in a selected region. Findings will be reported.

I. INTRODUCTION

Density variations in the ocean medium influence the acoustic intensity; thus, techniques must be developed to handle these density variations adequately. A number of techniques are in existence for such treatment. In this paper we enhance the numerical technique developed by Lee et al. [1] for handling vertical density variations to handle density variations horizontally. The purpose of this enhancement is to examine the effect caused by both vertical and horizontal variations. It is generally believed that the density variation in the vertical direction has a stronger influence on acoustic intensity than the density variations in the horizontal directions. A mathematical model having density variations in all directions is presented. A numerical solution follows. This numerical solution is incorporated into an existing 3-dimensional computer code FOR3D [2]. This updated code is applied to examine the overall density variations in a selected region whose inputs were obtained from Harvard University [3,4]. We report our findings based on this set of data in a selected region. Some discussions are given at the end.

II. A MATHEMATICAL MODEL AND ITS NUMERICAL SOLUTION

A 3-dimensional mathematical model [1] having vertical density variations has the expression

$$u_r = (-ik_o + ik_o[1 + \frac{1}{2} X^+ - \frac{1}{8}(X^+)^2 + \frac{1}{2} Y])u, \qquad (1)$$

where

$$X^+ = n^2(r,\theta,z) -1 + \frac{1}{k_o^2} [\rho(z)\frac{\partial}{\partial z} (\frac{1}{\rho(z)} \frac{\partial}{\partial z})], \qquad (2)$$

$$Y = \frac{1}{k_o^2 r^2} \frac{\partial^2}{\partial \theta^2} . \qquad (3)$$

A complete mathematical model to accommodate density variations in all directions must contain the capability of handling the horizontal density variation. A mathematical model, based upon Ref. 1, is developed to handle horizontal density variations which has the expression

$$u_r = (-ik_o + ik_o[1 + \frac{1}{2} X^+ - \frac{1}{8}(X^+)^2 + \frac{1}{2} Y^+])u, \qquad (4)$$

where

$$Y^+ = \frac{1}{k_o^2 r^2} \rho(\theta,z) \frac{\partial}{\partial \theta} (\frac{1}{\rho(\theta,z)} \frac{\partial}{\partial \theta}). \qquad (5)$$

This formulation allows Eq.(4) to be solved by the same numerical procedure used to solve Eq.(1). As a result, Eq.(4) can be solved in two steps by 2 tri-diagonal systems of equations, provided both vertical and horizontal density variations are treated satisfactorily on the interface. Vertical and horizontal density variations are accounted for in steps 1 and 2, respectively. Detailed numerical solutions of Eq.(1) as well as the numerical density treatment can be found in Ref. 1.

III. AN APPLICATION

An environment was selected from a Gulfcast generated by the Harvard Open Ocean Model [3,4]. A 25 Hz source was placed 100 meters in depth above the continental slope at latitude 39.5°N, and longitude 72°W. Propagation loss predictions were computed in a 3-dimensional wedge that measured 10° in width. Direction of propagation of the center ray emanating from the source located at the vertex of the wedge was 180°T. The water depth in the wedge is variable and ranges from approximately 500 meters to approximately 3500 meters. Density in the water and in the bottom are assumed to be 1.0 g/cm^3 and 1.8 g/cm^3, respectively. Four propagation loss predictions computed by the FOR3D model are shown in Fig. 1. Since FOR3D computes propagation loss in two steps, it was possible to modify the code such that different densities are accounted for in steps 1 and 2. In step 1, density variations in the vertical plane are accounted for resulting in a 2-dimensional solution. In step 2, azimuthal density variations in the horizontal plane are accounted for resulting in a 3-dimensional solution. The following table summarizes the various combinations of density conditions under which these solutions were computed.

			Step 1	Step 2
Curve	Water Density	Bottom Density	Vertical Density	Horizontal Density
1	1.0 g/cm^3	1.0 g/cm^3	1.0 g/cm^3	1.0 g/cm^3
2	1.0 g/cm^3	1.8 g/cm^3	1.8 g/cm^3	1.8 g/cm^3
3	1.0 g/cm^3	1.8 g/cm^3	1.8 g/cm^3	1.0 g/cm^3
4	1.0 g/cm^3	1.8 g/cm^3	1.0 g/cm^3	1.8 g/cm^3

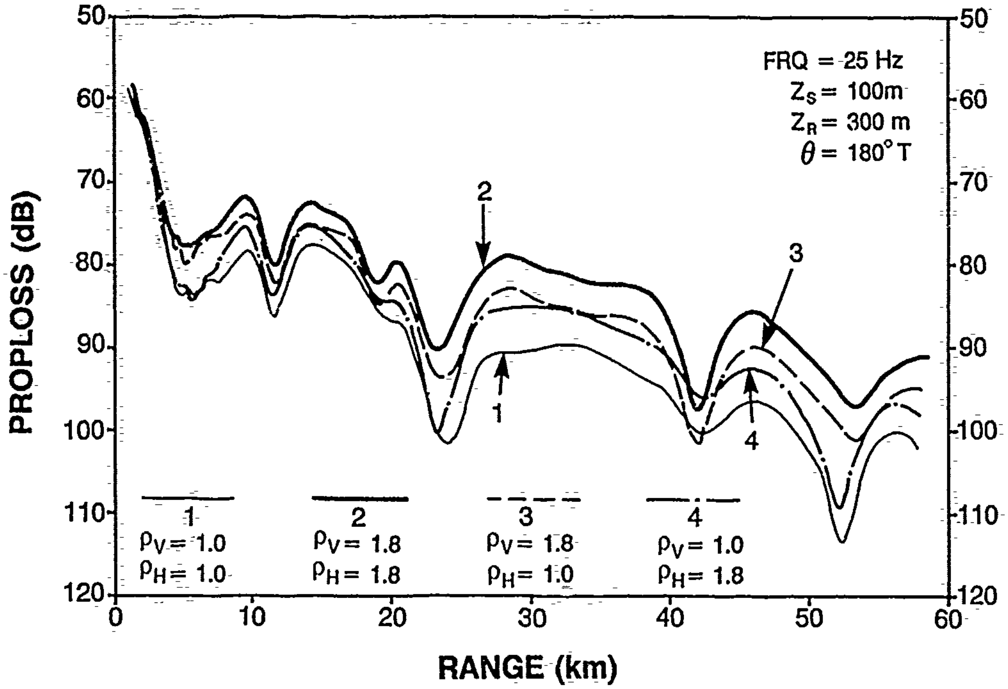


Fig. 1: Propagation loss with vertical and horizontal density variations

Curve 1 represents the solution at 300 meters in depth when density is set to 1.0 g/cm^3 everywhere; that is, there is no density variation. For curve 2, density in the bottom was set to 1.8 g/cm^3 and correctly accounted for in both steps 1 and 2. In curve 3, horizontal density variations were deliberately not accounted for; that is, no azimuthal density variation was detected by FOR3D when passing from water into a seamount. For curve 4, vertical density variations were not accounted for. A comparison of curve 1 (no density variations) with curve 2 (with density variations) shows differences of as much as 10 dB. Comparing curve 3 with 4 shows that the average level of propagation loss is midway between curves 1 and 2 and approximately the same implying that neither the vertical nor the horizontal treatment of density dominates in the solution. How ever, by comparing curves 1 with 4 and 2 with 3, it can be seen that the shape of the curve is influenced by the density variations in the vertical plane.

IV. CONCLUSIONS

The results show that, in general, neither the vertical nor the horizontal treatment of density variations dominate in the prediction of propagation loss. Both contribute equally well to the solution and must be accounted for. However, the shape of the propagation loss curve is determined by the density variations in the vertical plane.

ACKNOWLEDGMENT

This research was supported jointly by ONR Research Grant N00014-91-WX-24038 and Naval Underwater Systems Center Independent Research Project A65025.

REFERENCES

1. Lee, D., M. H. Schultz, and Y. Saad, "A three-dimensional wide angle wave equation with vertical density variations," in COMPUTATIONAL ACOUSTICS: Ocean-Acoustic Models and Supercomputing, eds. D. Lee, A. Cakmak, and R. Vichnevetsky, North-Holland, Amsterdam, 1990, 143-154.
2. Botseas, G., D. Lee, and D. King, "FOR3D: A computer model for solving the LSS three-dimensional wide angle wave equation," Technical Report 7943, Naval Underwater Systems Center, New London, CT, USA, 1987.
3. Robinson, A. R. and L. J. Walstad, "The Harvard Open Ocean Model: Calibration and application to dynamical process, forecasting, and data assimilation studies," J. Applied Numerical Mathematics, Vol. 3, 1987, 89-131.
4. Glenn, S. M. and A. R. Robinson, "Nowcasting and forecasting of oceanic dynamic and acoustic fields, in COMPUTATIONAL ACOUSTICS: Ocean-Acoustic Models and Supercomputing, eds. D.Lee, A.Cakmak, and R.Vichnevetsky, North-Holland, Amsterdam, 1990, 117-128.

HARMONIC ANALYSIS IN PHASE SPACE-APPLICATIONS TO DIRECT AND INVERSE WAVE PROPAGATION

LOUIS FISHMAN
Department of Mathematical
and Computer Sciences
Colorado School of Mines
Golden, CO 80401 USA

AND

RONALD I. BRENT
Department of Mathematics
University of Lowell
Lowell, MA 01854 USA

Abstract – The application of phase space and functional integal methods to scalar and vector, direct and inverse, wave propagation problems is briefly outlined.

I. INTRODUCTION

This paper addresses wave propagation in extended, inhomogeneous, multidimensional environments capable of channeling energy over many wavelengths. Sound propagation in the ocean and electromagnetic guided-wave propagation are two examples. For the most part, the application of classical, "macroscopic" methods has resulted in direct wave field approximations, derivations of approximate wave equations, and discrete numerical approximations. In the last several decades, however, developments in Fourier analysis, partial differential equations, mathematical physics, among others have been synthesized into what is now called harmonic analysis in phase space [FO]. This analysis on the configuration space R^n done by working in the phase space $R^n \times R^n$ has produced sharp, "microscopic" tools (pseudo differential and Fourier integral operators, wave packets) appropriate for attacking wave propagation problems in extended environments [FO]. In conjunction with the global functional integral techniques [SC] pioneered by Wiener (Brownian motion) and Feynman (quantum mechanics), and so successfully applied today in quantum field theory and statistical physics, the n-dimensional wave field propagators can be both represented explicitly and computed directly. The phase space, or "microscopic," methods and path (functional) integral representations provide the appropriate framework to extend homogeneous Fourier methods to extended inhomogeneous environments, in addition to suggesting the basis for the formulation and solution of corresponding arbitrary-dimensional nonlinear inverse problems [WE].

II. WAVE EQUATION MODELING AND FORMAL WAVE FIELD SPLITTING

For sound propagation in the ocean, the initial modeling is provided by the n-dimensional scalar Helmholtz equation,

$$\left(\nabla^2 + \bar{k}^2 K^2(\underline{x})\right)\phi(\underline{x}) = 0 \ , \tag{1}$$

where $K(\underline{x})$ is the refractive index field and $\bar{k}$ is a reference wave number. The environment can be characterized by a refractive index field with a compact region of arbitrary (n dimensional) variability superimposed upon a transversely inhomogeneous ($(n$-1)-dimensional) background profile. Splitting the wave field $\phi(\underline{x})$ into two components, $\phi^+(\underline{x})$ and $\phi^-(\underline{x})$, via the transformation

$$\begin{pmatrix} \phi(\underline{x}) \\ \partial_x\phi(\underline{x}) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ i\mathrm{B}_1 & i\mathrm{B}_2 \end{pmatrix} \begin{pmatrix} \phi^+(\underline{x}) \\ \phi^-(\underline{x}) \end{pmatrix} \ , \tag{2}$$

where B_1 and B_2 are the two operator solutions of the simple quadratic operator equation

$$\mathrm{B}^2 - \left(\bar{k}^2 K^2(\underline{x}) + \nabla_t^2\right) = 0 \ , \tag{3}$$

results in the equivalent formulation [WE]

$$\partial_x \begin{pmatrix} \phi^+(\underline{x}) \\ \phi^-(\underline{x}) \end{pmatrix} = \tag{4}$$

$$\begin{pmatrix} \left(-\tfrac{i}{2}\partial_x \mathrm{B}_1^{-1} + 1\right) i\mathrm{B}_1 & \left(\tfrac{i}{2}\partial_x \mathrm{B}_2^{-1}\right) i\mathrm{B}_2 \\ \left(\tfrac{i}{2}\partial_x \mathrm{B}_1^{-1}\right) i\mathrm{B}_1 & \left(-\tfrac{i}{2}\partial_x \mathrm{B}_2^{-1} + 1\right) i\mathrm{B}_2 \end{pmatrix} \begin{pmatrix} \phi^+(\underline{x}) \\ \phi^-(\underline{x}) \end{pmatrix} .$$

It is straightforward to show that (1) – (3) imply (4) and, conversely, that (2) – (4) imply (1). Moreover, (2) and (4) are consistent. Choosing B_1 to correspond to the forward (outgoing) wave radiation condition and B_2 to correspond to the backward wave radiation condition completes the identification.

Sound propagation in the ocean, certain guided-wave electromagnetic propagation problems, and borehole-to-borehole seismic modeling are near one-way propagation problems in extended environments. To zeroth-order, these propagation problems can be viewed as transversely inhomogeneous, suggesting a weak-backscatter perturbation approach to the general direct and inverse wave propagation problems. In this context, the transversely inhomogeneous problem is first solved and then used to attack the more general formulation [WE]. Mathematically, this is expressed through the approximate diagonalization of the first-order Helmholtz system in (4). For a transversely inhomogeneous environment, $K^2(\underline{x}) = K^2(\underline{x}_t)$, (4) is diagonal. The forward evolution (one-way) equation,

$$(i/\bar{k})\,\partial_x\phi^+(x,\underline{x}_t) + \left(K^2(\underline{x}_t) + (1/\bar{k}^2)\nabla_t^2\right)^{1/2}\phi^+(x,\underline{x}_t) = 0 \ , \tag{5}$$

is the formally exact wave equation for propagation in a transversely inhomogeneous half-space supplemented with appropriate outgoing wave radiation and initial-value conditions [WE].

Unlike the transversely inhomogeneous scalar Helmholtz equation, which factors in terms of a formal square root operator, the transversely inhomogeneous Maxwell's equations do not admit such a simple decomposition. The frequency-domain form of Maxwell's equations in three spatial dimensions is taken to be

$$\partial_x^2 \underline{E}(\underline{x}) + \underline{\underline{C}}(\underline{x})\cdot\partial_x\underline{E}(\underline{x}) + \underline{\underline{A}}^2(\underline{x})\cdot\underline{E}(\underline{x}) = 0 \ , \tag{6}$$

where, under transversely inhomogeneous conditions, the matrix operators $\underline{\underline{A}}^2(\underline{x}_t)$ and $\underline{\underline{C}}(\underline{x}_t)$ are defined in [BR]. For transversely inhomogeneous environments, the electric field vector ($\underline{E}$) can be split into physical forward ($\underline{E}^+$) and backward ($\underline{E}^-$) propagating wave field components. The exact diagonalization is given by [BR]

$$\partial_x \begin{pmatrix} \underline{E}^+(\underline{x}) \\ \underline{E}^-(\underline{x}) \end{pmatrix} = \begin{pmatrix} i\underline{\underline{B}}_1 & 0 \\ 0 & i\underline{\underline{B}}_2 \end{pmatrix} \begin{pmatrix} \underline{E}^+(\underline{x}) \\ \underline{E}^-(\underline{x}) \end{pmatrix} \ , \tag{7}$$

where

$$\begin{pmatrix} \underline{E}(\underline{x}) \\ \partial_x\underline{E}(\underline{x}) \end{pmatrix} = \begin{pmatrix} \underline{\underline{I}} & \underline{\underline{I}} \\ i\underline{\underline{B}}_1 & i\underline{\underline{B}}_2 \end{pmatrix} \begin{pmatrix} \underline{E}^+(\underline{x}) \\ \underline{E}^-(\underline{x}) \end{pmatrix} \ , \tag{8}$$

$\underline{\underline{I}}$ is the 3 x 3 identity matrix, and $\underline{\underline{B}}_1$ and $\underline{\underline{B}}_2$ are the two matrix operator solutions of the generalized quadratic operator equation

$$\underline{\underline{B}}^2 - i\underline{\underline{C}}(\underline{x}_t)\cdot\underline{\underline{B}} - \underline{\underline{A}}^2(\underline{x}_t) = 0 \ . \tag{9}$$

For a transversely-inhomogeneous environment, (6) is equivalent to (7) [BR].

III. PHASE SPACE AND PATH INTEGRAL CONSTRUCTIONS

The formal one-way Helmholtz wave equation (5) can be recast and analyzed within the phase space framework as a Weyl pseudo-differential equation in the form [WE]

$$(i/\bar{k})\partial_x\phi^+(x,\underline{x}_t)+(\bar{k}/2\pi)^{n-1}\int_{R^{2n-2}} d\underline{x}'d\underline{p}_t$$

$$\Omega_{\mathrm{B}}\left(\underline{p}_t,(\underline{x}_t+\underline{x}_t')/2\right)\exp\left(i\bar{k}\underline{p}_t\ \ (\underline{x}_t-\underline{x}_t')\right)\phi^+(x,\underline{x}_t')=0\ \ , \quad (10)$$

where $\Omega_{\mathrm{B}}(\underline{p},\underline{q})$ is the symbol associated with the square root Helmholtz operator $\mathrm{B} = \left(K^2(\underline{q})+(1/\bar{k}^2)\nabla_{\underline{q}}^2\right)^{1/2}$. In the Weyl pseudo-differential-operator calculus, the operator symbol $\Omega_{\mathrm{B}}(\underline{p},\underline{q})$ is defined through the Weyl composition equation

$$\Omega_{\mathrm{B}^2}(\underline{p},\underline{q}) \ = \ K^2(\underline{q})-\underline{p}^2 =$$

$$(\bar{k}/\pi)^{2n-2}\int_{R^{4n-4}} d\underline{t}d\underline{x}d\underline{y}d\underline{z}\,\Omega_{\mathrm{B}}(\underline{t}+\underline{p},\underline{x}+\underline{q})$$

$$\cdot\,\Omega_{\mathrm{B}}(\underline{y}+\underline{p},\underline{z}+\underline{q})\exp\left(2i\bar{k}(\underline{x}\cdot\underline{y}-\underline{t}\cdot\underline{z})\right)\ \ . \quad (11)$$

with $\Omega_{\mathrm{B}^2}(\underline{p},\underline{q})$ the symbol associated with the square of B, $\mathrm{B}^2 = \left(K^2(\underline{q})+(1/\bar{k}^2)\nabla_{\underline{q}}^2\right)$ [WE].

Solution representations for pseudo-differential equations such as (10) can be directly expressed in terms of infinite-dimensional functional, or path, integrals [SC], following from the Markov, or semigroup, property of the propagator. The path integral representation for the propagator takes the form [WE]

$$G^+(x,\underline{x}_t|0,\underline{x}_t') \ = \ \lim_{N\to\infty}\int_{R^{(n-1)(2N-1)}}\prod_{j=1}^{N-1}d\underline{x}_{jt}\prod_{j=1}^{N}(\bar{k}/2\pi)^{n-1}d\underline{p}_{jt}$$

$$\cdot\,\exp\Bigg(i\bar{k}\sum_{j=1}^{N}\Big(\underline{p}_{jt}\cdot(\underline{x}_{jt}-\underline{x}_{j-1t})$$

$$+(x/N)H(\underline{p}_{jt},\underline{x}_{jt},\underline{x}_{j-1t})\Big)\Bigg) \quad (12)$$

where $H(\underline{p},\underline{q}'',\underline{q}')$ is related to the standard pseudo-differential operator symbol [WE].

The one-way marching algorithm is based on (1) the marching range step (following from the path integral), (2) a sophisticated symbol analysis (reflecting the detailed study of the (Helmholtz) Weyl composition equation (11)), and (3) Fourier component, or wave number, filtering in phase space (for increased efficiency, decreased computational time, and reduced error). The detailed numerical algorithm is discussed in [WE]. Sufficiently accurate approximations of the square root ΨDO symbol over the relevant region of phase space result in very accurate numerical wave field calculations [WE].

The phase space and path integral constructions for the one way Helmholtz equation (5) can be extended to the one-way Maxwell equation (7). The explicit phase space constructions are particularly important in this case since the propagation operator is defined through a generalized quadratic operator equation rather than in a simple manner in terms of formal square root operators. In analogy with (10), the one-way equation (7) takes the form [BR]

$$(i/\bar{k})\partial_x\underline{E}^+(x,\underline{x}_t)+(\bar{k}/2\pi)^2\int_{R^4} d\underline{x}_t'd\underline{p}_t\exp\left(i\bar{k}\underline{p}_t\cdot(\underline{x}_t-\underline{x}_t')\right)$$

$$\cdot\,\underline{\underline{\Omega}}_{\underline{\underline{B}}}\left(\underline{p}_t,(\underline{x}_t+\underline{x}_t')/2\right)\cdot\underline{E}^+(x,\underline{x}_t')=0\,, \quad (13)$$

with the corresponding composition equation given by

$$\underline{\underline{\Omega}}_{\underline{\underline{A}}^2}(\underline{p},\underline{q}) \ = \ (\bar{k}/\pi)^4\int_{R^8} d\underline{t}d\underline{x}d\underline{y}d\underline{z}$$

$$\cdot\left(\underline{\underline{\Omega}}_{\underline{\underline{B}}}(\underline{t}+\underline{p},\underline{x}+\underline{q})-(i/\bar{k})\underline{\underline{C}}(\underline{x}+\underline{q})\right)$$

$$\cdot\,\underline{\underline{\Omega}}_{\underline{\underline{B}}}(\underline{y}+\underline{p},\underline{z}+\underline{q})\exp\left(2i\bar{k}(\underline{x}\cdot\underline{y}-\underline{t}\cdot\underline{z})\right)\ \ . \quad (14)$$

In (14), $\underline{\underline{\Omega}}_{\underline{\underline{A}}^2}(\underline{p},\underline{q})$ is defined in [BR], $\underline{p}=(p_1,p_2)$, $\underline{q}=(q_1,q_2)$. and the solution $\underline{\underline{\Omega}}_{\underline{\underline{B}}}(\underline{p},\underline{q})$ corresponding to the outgoing (forward) radiation condition is chosen. The operator symbol in the scalar case has been replaced by an operator symbol matrix in the vector case. Analogous path integral and marching algorithm constructions follow [BR].

For wave propagation problems in the presence of two (generally different) transversely inhomogeneous half-spaces separated by a planar transition region of arbitrary length and inhomogeneity, the one-way phase space and path integral methods can be combined with invariant imbedding techniques. For the Helmholtz equation, designating the left and right boundaries of the transition region at $x = a$ and $x = b$, respectively, and generally locating a source in each half-space, the incident wave fields are connected to the scattered wave fields through the operator-valued scattering matrix $\underline{\mathrm{S}}(a,b)$,

$$\begin{pmatrix}\phi^+(b,\underline{x}_t)\\ \phi^-(a,\underline{x}_t)\end{pmatrix} \ = \ \underline{\mathrm{S}}(a,b)\cdot\begin{pmatrix}\phi^+(a,\underline{x}_t)\\ \phi^-(b,\underline{x}_t)\end{pmatrix} \quad (15)$$

$$= \begin{pmatrix}\mathrm{T}^+(a,b) & \mathrm{R}^-(a,b)\\ \mathrm{R}^+(a,b) & \mathrm{T}^-(a,b)\end{pmatrix}\cdot\begin{pmatrix}\phi^+(a,\underline{x}_t)\\ \phi^-(b,\underline{x}_t)\end{pmatrix}.$$

The scattering matrix is defined in terms of the appropriate forward (right-traveling) and backward (left-traveling) reflection and transmission operators associated with the transition region. Invariant imbedding [WE] intuitively views the scattering matrix for a finite region as being composed of scattering matrices of a large number of contiguous subregions, and thus computes the effect of adjoining a very thin slab to the right-hand side of the transition region. The resulting coupled invariant imbedding initial-value system is [WE]

$$\partial_b\underline{\mathrm{S}}(a,b) \ = \ \begin{pmatrix}1 & \mathrm{R}^-(a,b)\\ 0 & \mathrm{T}^-(a,b)\end{pmatrix}\cdot\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

$$\cdot\,\underline{\mathrm{H}}(b)\cdot\begin{pmatrix}\mathrm{T}^+(a,b) & \mathrm{R}^-(a,b)\\ 0 & 1\end{pmatrix}. \quad (16)$$

In (16), $\underline{\mathrm{H}}(b)$ is the operator-valued matrix in (4) evaluated at $x = b$. The initial conditions for (16) are determined by the appropriate planar interface problem at $b = a$. The invariant imbedding procedure transforms the Helmholtz boundary-value problem of (4) into an initial-value problem through the complete specification of the two-component wave field column vector at either $x = a$ or $x = b$. The formal operators in (4), (15), and (16) are explicitly represented as Weyl pseudo-differential operators [WE]. Equation (16) provides the basis for both direct and inverse algorithms [WE].

ACKNOWLEDGMENTS

This work was supported by NSF, AFOSR, and ONR.

REFERENCES

[BR] R.J. Brent and L. Fishman, *Phase space factorization analysis for vector electromagnetic wave propagation*, JOSA A, submitted for publication (1991).

[FO] G.B. Folland, *Harmonic Analysis in Phase Space*, Princeton University Press, Princeton, 1989.

[SC] L.S. Schulman, *Techniques and Applications of Path Integration*, Wiley, New York, 1981

[WE] V.H. Weston, J.P. Corones, L. Fishman, and J.J. McCoy, *Wave Splitting with Applications to Wave Propagation and Inverse Scattering*, SIAM, Philadelphia, 1991.

NUMERICAL SOLUTIONS OF THE ELASTIC WAVE EQUATIONS

G. H. Knightly[*], G.-Q. Li, D. F. St.Mary[†]
Center for Applied Mathematics and Mathematical Computation
Department of Mathematics
University of Massachusetts
Amherst, MA 01003, USA

Ding Lee[‡]
Naval Underwater Systems Center
New London, CT 06320, USA

Abstract The question of the utilization of parabolic equations to approximate the elastic wave equations with liquid/solid interface is studied. The main consideration is the development of numerical schemes which are stable. Crank Nicolson discretization schemes for a parabolic elastic system of equations with interface are discussed. Two approaches to the solution of the system are considered, in one the full problem is discretized directly, in the other the solid and liquid parts are solved successively.

1 Introduction

Recently, considerable attention has been given to questions revolving around the effects of the ocean bottom interface on propagation in the water column. In fact, the Office of Naval Research, USA, is currently supporting fundamental research in this area through an accelerated research initiative called *Acoustics Reverberation Special Research Program*. Here, the interest is in scatter from the ocean bottom, the partitioning of the incident acoustic wavefield by the bottom/subbottom, and re-radiation into the water column.

Parabolic approximation methods, having been applied extensively, and successfully, to underwater acoustics problems in the water column are beginning to be utilized in the case of an elastic medium, namely, the ocean bottom, see e.g. [6, 7, 3, 1, 2]. This activity usually involves the approximation of the elastic wave equations by a parabolic-type partial differential equation and the implementation of interface conditions to represent the liquid/solid interface under consideration. In [6], Shang & Lee chose to focus on the implementation of this interface in 2-D by working to connect the traditional parabolic equation, (PE), which is applied in the liquid medium, to a parabolic approximation of the elastic wave equations in the elastic medium. The scheme developed in [6] is unstable. In this paper, we reconsider the approach in [6], with the goal of creating a numerical scheme with robust stability properties. The essence of the approach in [6] involves the utilization of a parabolic approximation to the elastic wave equations which Shang & Lee extract from the work of McCoy [4], a clever Taylor polynomial approximation to a second derivative gleaned from [5], and three systems of equations – one representing the liquid medium, an interface system of three equations in three unknowns, and a system representing the elastic medium. In the development presented here, two approaches to the solution of the system are considered, in one the full problem is discretized directly, in the other the solid and liquid parts are solved successively. The Crank-Nicolson discretization method is employed in the case of all equations in an attempt to ensure stability.

In the next section we present the system of equations which constitutes the parabolic approximation to the elastic wave equation with interface. In section three we describe the Crank-Nicolson discretization of the system and suggest an approach to breaking it into two systems to be solved successively.

2 BACKGROUND

We adopt the notation of [6] and refer the reader to [6] for a detailed description of the derivation of the material presented in this section. The parabolic approximating equations utilized here emanate from potential equations in both media. In the elastic medium the potential equations might be thought of as a series of "local" potential equations (the medium is said to be *locally* isotropic) which have been joined so as to have nonconstant parameter functions. In the liquid medium the parabolic approximation process is well established and for these purposes yields the standard (PE)

$$\begin{aligned} \frac{\partial A_1}{\partial r} &= a_1 A_1 + b_1 \frac{\partial^2 A_1}{\partial z^2}, \\ a_1 &= \frac{i}{2k_0}\left[k_1^2(r,z) - k_0^2\right], \quad b_1 = \frac{i}{2k_0}, \end{aligned} \tag{2.1}$$

where k_0 is a reference wave number. The parabolic system approximating the elastic wave equations is given by

$$\begin{aligned} \frac{\partial A_2}{\partial r} &= a_2 A_2 + b_2 \frac{\partial^2 A_2}{\partial z^2} + c_2 \frac{\partial B_2}{\partial z} \\ \frac{\partial B_2}{\partial r} &= a_2' B_2 + b_2' \frac{\partial^2 B_2}{\partial z^2} + c_2' \frac{\partial A_2}{\partial z}, \end{aligned} \tag{2.2}$$

where c_2 and c_2' are coupling coefficients whose definitions along with other symbols are given by

$$\begin{aligned} a_2 &= (i/2\bar{k}_D)[k_D^2(r,z) - \bar{k}_D^2], \quad b_2 = i/2\bar{k}_D, \\ c_2 &= (-1/2\bar{k}_D)(\Delta\bar{k}\bar{\varepsilon}_{\mu\rho}), \\ a_2' &= (i/2\bar{k}_S)[k_S^2(r,z) - \bar{k}_S^2], \quad b_2' = 1/2\bar{k}_S, \\ c_2' &= (-1/2\bar{k}_S)(\Delta\bar{k}\bar{\varepsilon}_{\mu\rho}^*), \\ \bar{\varepsilon}_{\mu\rho} &= \frac{1}{\Delta r}\int_{n\Delta r}^{(n+1)\Delta r} \varepsilon_{\mu\rho}(x,z)e^{i\Delta\bar{k}x}\,dx, \end{aligned}$$

[*] Work supported by ONR, N00014-90-J-1031
[†] Work supported by ONR, N00014-91-J-1119
[‡] Work supported by ONR, and NUSC Independent Research

where $\varepsilon_{\mu\rho} = 2(\overline{k}_D/\overline{k}_S)\varepsilon_\mu - \varepsilon_\rho$, $\Delta\overline{k} = \overline{k}_S - \overline{k}_D$, and * denotes complex conjugate. Now

$$\overline{k}_D^2 = \omega^2/\overline{c}_D^2 = \overline{\rho}_2\omega^2/(\overline{\lambda}_2 + 2\overline{\mu}_2),$$

and

$$\overline{k}_S^2 = \omega^2/c_S^2 = \overline{\rho}_2\omega^2/\overline{\mu}_2,$$

where λ, and μ are Lamé parameters, ω frequency, and additionally, e. g., $\overline{\lambda}$ represents an approximation to $\lambda(r,z)$ with relative error ϵ_λ,

$$\lambda_2 = \overline{\lambda}_2[1 + \varepsilon_\lambda(r,z)],$$

over an interval of length Δr.

Again, the derivation of the inteiface conditions in terms of the parabolic potentials is contained in [6]. We re-piesent them here for purposes of completeness. We would also like to call attention to the fact that in their derivation the usual PE terms are dropped, namely, terms of the form $-\frac{\partial^2()}{\partial r^2}$, $\frac{1}{r}()$ The continuity of vertical components of displacement and of stress translate respectively to

$$\frac{\partial A_1}{\partial z} = K_D\frac{\partial A_2}{\partial z} + K_S\left(\frac{2B_2}{\partial r} + i\overline{k}_S B_2\right), \tag{2.3a}$$

$$\begin{gathered}-\rho_1\omega^2 A_1 = 2\mu_2\left(\frac{\partial^2 B_2}{\partial r\partial z} + i\overline{k}_S\frac{\partial B_2}{\partial z}\right)K_s \\ + \left(\lambda_2\left(\frac{\partial^2 A_2}{\partial z^2} + 2i\overline{k}_D\frac{\partial A_2}{\partial r} - \overline{k}_D^2 A_2\right) + 2\mu_2\frac{\partial^2 A_2}{\partial z^2}\right)K_D.\end{gathered} \tag{2.3b}$$

Finally, the vanishing of the horizontal components of stress on the interface yields

$$2\frac{\partial^2 A_2}{\partial z\partial r} + 2i\overline{k}_D\frac{\partial A_2}{\partial z} = \left(\frac{\partial^2 B_2}{\partial z^2} - 2i\overline{k}_S\frac{\partial B_2}{\partial r} + \overline{k}_S^2 B_2\right)K_{SD} \tag{2.3c}$$

where

$$K_D = \sqrt{\frac{k_0}{\overline{k}_D}}\,e^{i\Delta_D r}, K_S = \sqrt{\frac{k_0}{\overline{k}_s}}\,e^{i\Delta_S r},$$

$$K_{SD} = K_{SD}(r) = \sqrt{\frac{\overline{k}_D}{\overline{k}_S}}\,e^{i\Delta_{SD} r},$$

$\Delta_D = \overline{k}_D - k_0$, $\Delta_S = \overline{k}_S - k_0$, and $\Delta_{SD} = \overline{k}_S - \overline{k}_D$,

3 DISCRETIZATION

The system of equations (2.1), (2.2), (2.3a-c) constitutes the parabolic elastic potential system with interface. We remark that each of the three equations (2.3a-c) is considered to hold on the interface. In particular, these equations represent the "slip" interface condition and thus we shall have a value for each of the unknown functions A_1, A_2, and B_2 on the interface. We shall perform a Crank-Nicolson discretization of each of the equations in the system. The resulting discrete system which includes all of the unknowns is a very large system. One solution approach to this aspect of the over-all problem is to attempt to divide the problem along the natural boundary of the liquid/solid interface.

The equations (2.3a-c) require special attention since A_1 is defined only for values of z on or above the interface, and A_2, B_2, only for values on or below it. In particular, second partial derivatives require extraordinary measures. Several possibilities suggest themselves: i) to replace the second order partial derivatives in z in (2.3b-c) by using the appropriate approximating partial differential equation in (2.2) at the interface points, e. g. in (2.3b) $\partial^2 A_2/\partial z^2$ is replaced by

$$\frac{1}{b_2}\left(\frac{\partial A_2}{\partial r} - a_2 A_2 - c_2\frac{\partial B_2}{\partial z}\right)_j,$$

where j represents the interface level, ii) to approximate these second order z-partial derivatives by a Taylor polynomial, e. g.

$$\left(\frac{\partial^2 A_2}{\partial z^2}\right)_j \approx -\frac{2}{\Delta z}\left(\frac{\partial A_2}{\partial z}\right)_j + \frac{2}{\Delta z^2}\left((A_2)_{j+1} - (A_2)_j\right),$$

iii) to use the value of the second derivative at a lower node.

It would be desirable to develoo a stability analysis and/or computer implementation of ea [illegible] these approaches to the "full" system. Our efforts have [illegible] concentrated to date on dividing the full system into two parts along the interface. Generally, we have included the discretized (2.3a) in the liquid subsystem, and discretizations of the modified (2.3b-c) equations in the solid subsystem. In attempting to accomplish the dissociation into subsystems somewhat arbitrary decisions are made, e.g. in solving the subsystem for A_2, B_2 at the $(n+1)^{th}$ range step, certain values of A_1 are required at the $(n+1)^{th}$ range step, thus as a consequence of breaking the system into two subsystems we replace these values by the corresponding values at the n^{th} range step. Methods i) and ii) have been implemented in the context just described and seem to give reasonable agreement with an exact solution of a test problem presented in [6] over the first few hundred iterations. Residuals eventually become unacceptably large as the number of iterations continue. Methods related to iii) remain a subject for future research.

References

[1] M. D. Collins, A Higher-Order Parabolic Equation for Wave Propagation in an Ocean Overlying an Elastic Bottom, *J. Acoust. Soc. Am.* 86, 1459-1464 (1989).

[2] M. D. Collins, Higher-Order Padé Approximations for Accurate and Stable Elastic Parabolic Equations with Application to Interface Wave Propagation, *J. Acoust. Soc. Am.* 89, 1050-1057 (1991).

[3] R. R. Greene, The Rational Approximation to the Acoustic Wave Equation with Bottom Interaction, *J. Acoust. Soc. Am.* 76, 1764-1773 (1984).

[4] J. J. McCoy, A Parabolic Theory of Stress Wave Propagation through Inhomogeneous Linearly Elastic Solid, *J. Appl. Mech.*. 44, 462-468 (1977).

[5] S.T. McDaniel and D. Lee, A Finite-Difference Treatment of Interface Conditions for the Parabolic Wave Equation: The Horizontal Interface *J. Acoust. Soc. Am.* 71, 855-858 (1982).

[6] E. C. Shang and D. Lee, A Numerical Treatment of the Fluid /Elastic Interface under Range-Dependent Environments, *J. Acoust. Soc. Am.* 85, 654-660 (1989).

[7] B T R. Wetton and G. H. Brooke, One-Way Wave Equation for Seismoacoustic Propagation along Rough and Sloping Interfaces, *J Acoust. Soc. Am.* 87, 624-632 (1990).

SIMULATION OF ACOUSTIC PULSE PROPAGATION THROUGH A TURBULENT MEDIUM

Allan D. Pierce
Graduate Program in Acoustics and Department of Mechanical Engineering
157 Hammond Building
Pennsylvania State University
University Park, PA 16802 U.S.A.
Tel: (814) 865-3161; Fax: (814) 863-7222

Abstract – The improved simulation of sonic boom propagation through the real atmosphere requires greater understanding of how the transient acoustic pulses popularly termed sonic booms are affected by atmospheric turbulence. Two primary turbulence effects have been identified: (1) the thickening of the nominally abrupt shock at the beginning of the pulse waveform and (2) the spiking and rounding of the portion of the waveform that immediately follows the shock. The turbulence-induced thickening effect is typically larger than (although occasionally much weaker than) the thickening effect caused by molecular relaxation, which involves nonlinear effects. The present paper describes novel procedures to simulate these effects.

I. INTRODUCTION

Sonic boom distortion by turbulence has been considered by many authors (1 – 3); previous work includes attempts (generally regarded as successful) by Pierce (4) and by Crow (5, 6) to explain the random spiking and rounding of waveforms, although a complete and satisfactory statistical theory has not yet been achieved. The effects of turbulence on rise times were considered by Plotkin and George (7), Pierce (8), and by Ffowcs-Williams and Howe (9). None of the latter theories were regarded as wholly satisfactory, however, and several authors suggested that tuirbulence played a minor role in the rise time phenomenon and that the dominant mechanism was molecular relaxation. Recently, the author and his colleagues have discovered (10), after a somewhat careful comparison of data with theoretical predictions based on the molecular relaxation model, that that mechanism tends to underestimate actual rise times by factors of the order of three. Thus, the overall question of how turbulence affects sonic boom waveforms deserves further consideration.

II. SONIC BOOM WAVEFORMS

A typical sonic boom waveform, acoustic portion of pressure versus time, is shown in Fig. 1. Note that the peak pressure is of the order of 70 Pa and that the waveform duration

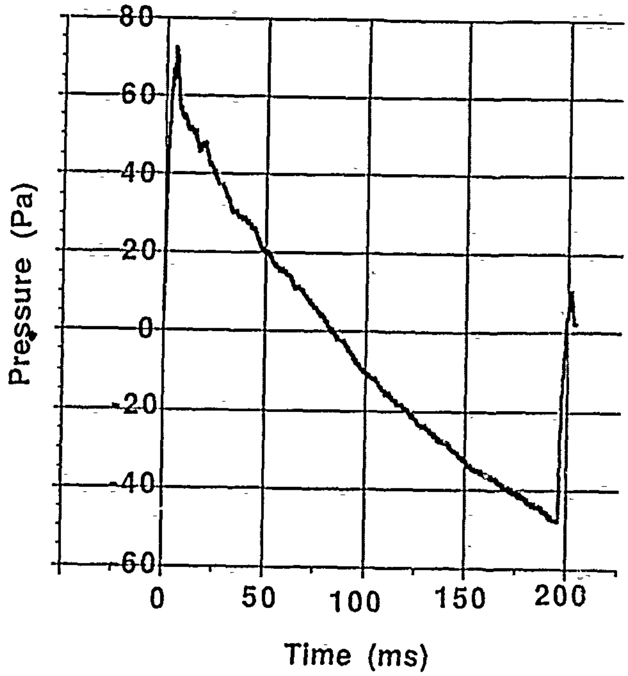


Fig. 1. Sonic boom waveform recorded at ground during overflight of an SR-71 at 66,000 ft altitude with speed of Mach 2.6.

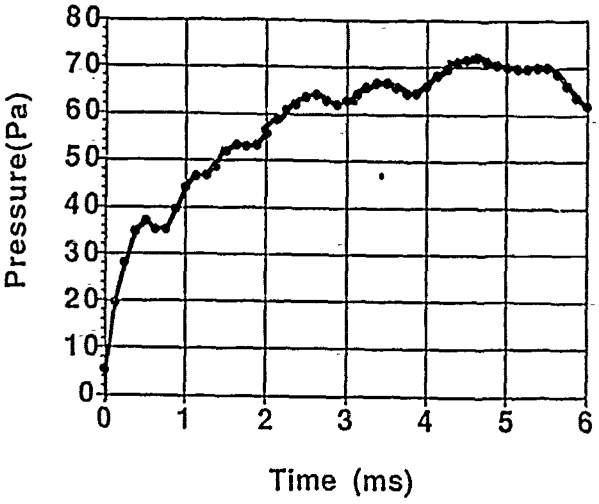


Fig. 2. Early portion (rise phase) of the waveform shown in Fig. 1.

is of the order of 200 ms. The standard idealization of such a waveform is that of an N-wave, an abrupt initial shock followed by a linear decrease of pressure through zero, terminated by a second abrupt shock. This particular waveform has one of the aberrations believed (4) to be caused by turbulence, in that there are small spike features near the leading and trailing edges of the shock, such that the jump in overpressure at the shocks is slightly higher than that of the overall N-shape that fits most of the waveform. Other aberrations commonly observed are rounded profiles where the initial shock is replaced by a much smaller jump followed by a slower transition of the duration of typically 10 to 15 ms up to the nominal N-shaped profile.

Figure 2 shows the first 6 ms of the waveform of Figure 1. On the scale with which this portion of the waveform is plotted, it is evident that the sudden increase in pressure corresponding to the shock is not abrupt and has a duration whose order of magnitude is 4 ms. This portion of the waveform, here termed the rise phase, is often characterized by a single number called the rise time, typically taken as that time for the pressure to rise from 10% to 90% of its peak value.

III. OUTLINE OF THE THEORY

A turbulent atmosphere may be characterized by a ambient fluid velocity v and sound speed c that vary from point to point. Techniques for synthesizing particular realizations (drawn from an ensemble) of such fields are described in a recent paper by Karweit et al. (11). The suggestion is made here that one do the synthesis is two stages. In the first stage, one leaves out all higher order wavenumber vector components beyond some cutoff wavenumber k_u, to the extent one is certain that geometrical acoustics will be very nearly wholly applicable for the propagation of the sonic boom from the aircraft trajectory to the considered observation point. The resulting medium is here called the background medium.

Given such a background medium, one identifies a central ray that propagates according to the ray tracing equations described in the author's book (12) and in a recent paper (13), and one describes distance along such a ray by a parameter s, and nominal time of arrival (ignoring nonlinear effects) of the leading shock by $\tau(s)$. The theory originally due to Hayes (14), and described in the author's text (12), applies for nonlinear propagation in the background medium along the central ray. When such a theory is modified to take molecular relaxation into account, one has (10):

$$p = B(s)g(s,t) \tag{1}$$

Here the amplitude function $B(s)$ continually adjusts to preserve the Blokhintzev invariant, in accord with the equation

$$\frac{d}{ds}\left\{\frac{B^2(s)|\mathbf{v}+c\mathbf{n}|(c+\mathbf{v}\cdot\mathbf{n})\delta A}{\rho c^3}\right\} = 0 \tag{2}$$

where δA is ray tube area. The quantity g satisfies the coupled equations

$$\frac{\partial g}{\partial t}+\left[(\frac{d\tau}{ds})^{-1}+\frac{d\mathcal{A}}{ds}(\frac{d\tau}{ds})^{-2}g\right]\frac{\partial g}{\partial s}-\delta\frac{\partial^2 g}{\partial s^2}+\sum_\nu(\Delta c)_\nu\frac{\partial g_\nu}{\partial s} = 0 \tag{3}$$

$$g_\nu + \tau_\nu\frac{\partial g_\nu}{\partial t} = \tau_\nu\frac{\partial g}{\partial t}. \tag{4}$$

The quantity

$$\mathcal{A}(s) = \int_0^s \frac{\beta B(s)\mathbf{e}_{\text{ray}}\cdot\mathbf{n}}{\rho c(c+\mathbf{v}\cdot\mathbf{n})^2}ds \tag{5}$$

is the age variable for the ray.

Lets ξ and η be two arbitrarily curvilinear coordinates such that in the vicinity of the central ray, the coordinates xi, η, and s, define an othogonal curvilinear coordinate system with metric tensor equal to unity along the central ray. To discover the diffraction correction to Eq. (3), one notes that for a homogeneous medium, in the absence of dissipation, relaxation, and nonlinear effects, and with an ambient fluid velocity in only the $+s$-direction, the quantity g satisfies the wave equation

$$c^2\left\{\frac{\partial^2 g}{\partial\xi^2}+\frac{\partial^2 g}{\partial\eta^2}\right\}-\left\{\frac{\partial}{\partial t}+(v+c)\frac{\partial}{\partial s}\right\}\left\{\frac{\partial}{\partial t}+(v-c)\frac{\partial}{\partial s}\right\}g = 0 \tag{6}$$

For a pulse propagating nearly in the $+s$ direction, this would simplify to

$$\left\{\frac{\partial}{\partial t}+(v+c)\frac{\partial}{\partial s}\right\}g+\frac{2}{c}\left\{\frac{\partial^2}{\partial\xi^2}+\frac{\partial^2}{\partial\eta^2}\right\}\int_0^s g\,ds = 0 \tag{7}$$

Consequently, the appropriate modification of Eq. (3) becomes

$$\begin{aligned}\frac{\partial g}{\partial t}+\left(\frac{d\tilde{\tau}}{ds}\right)^{-1}\frac{\partial g}{\partial s}+\frac{2}{c}\left\{\frac{\partial^2}{\partial\xi^2}+\frac{\partial^2}{\partial\eta^2}\right\}\int_0^s g\,ds \\ +\frac{d\mathcal{A}}{ds}(\frac{d\tau_c}{ds})^{-2}g\frac{\partial g}{\partial s}-\delta\frac{\partial^2 g}{\partial s^2}+\sum_\nu(\Delta c)_\nu\frac{\partial g_\nu}{\partial s} = 0\end{aligned} \tag{8}$$

where $\tilde{\tau}(\xi,\eta,s)$ is travel time [with the total turbulence with higher wavenumbers taken into account) along paths defined by the background medium. Readers may discern some similarities in this result with the parabolic equation, especially the formulation of Kriegsmann and Larsen (15), and with the nonlinear PE formulation of McDonald and Kuperman (16).

Acknowledgements – The research reported here was supported by NASA Langley Research Center. The author is particularly grateful to Drs. Gerry McAninch and Christine Darden for their encouragement and interest.

References – Space limitations preclude explicit listings of the references cited in the text.

ON THE COMPUTATIONAL FORMULA OF MODAL TRAVEL TIME PERTURBATION

E.C. Shang
CIRES, University of
Colorado/NOAA/WPL
Boulder,CO 80303,U.S.A.

and

Y.Y. Wang
CIRES, University of
Colorado/NOAA/WPL
Boulder,CO 80303,U.S.A.

Abstract-The computational formula of the adiabatic modal travel time perturbation has been derived. The linear constiuent and the nonlinear constituent are decomposed. Numerical results demonstrate that significant nonlinearity of modal travel time perturbation can be caused by strong oceanic mesoscale eddies.

1. THE EXACT FORMULA

The exact modal travel time perturbation is

$$\delta t_m = \int_0^R \left(\frac{1}{\tilde{U}_m} - \frac{1}{U_m}\right) dr \qquad (1)$$

where U_m and $\tilde{U}_m$ are modal group velocities corresponding to the unperturbed sound speed profile (SSP) $c_0(z)$ and the perturbed SSP $\tilde{c}(z)$, respectively. In Ref.[1], a rigorous formulation for calculating the modal group velocity has been given:

$$\frac{1}{U_m} = C_m \int_0^\infty \psi_m^2 \frac{1}{C^2}\, dz \qquad (2)$$

where c_m is the modal phase velocity and ψ_m is the normalized eigenfunction. Substituting eq.(2) into eq.(1), we get the exact modal travel time perturbation as follows:

$$\delta t_m = \int_0^R dr \left\{ \tilde{C}_m \int_0^\infty \frac{\tilde{\psi}_m^2}{\tilde{C}^2(z)} dz - C_m \int_0^\infty \frac{\psi_m^2}{C_0^2(z)} dz \right\} \qquad (3)$$

2. THE PERTURBATION FORMULA

The modal travel time perturbation can also be expressed by differential formulation according to the definition of modal group velocity:

$$\frac{1}{U_m} = \left(\frac{\partial}{\partial\omega} k_m\right)_{\omega_0} \qquad (4)$$

Then, the modal travel time perturbation is given by

$$\delta t_m = \int_0^R dr \left[\frac{\partial}{\partial\omega}(\tilde{k}_m - k_m)\right] \qquad (5)$$

The integrand of eq.(5) can be obtained from the eigen-value problem. The differential equation holding for unperturbed mode is:

$$\left[\frac{d^2}{dz^2} + k_0^2(z)\right]\psi_m = k_m^2 \psi_m \qquad (6)$$

and the differential equation holding for perturbed mode is:

$$\left[\frac{d^2}{dz^2} + \tilde{k}^2(z)\right]\tilde{\psi}_m = \tilde{k}_m^2 \tilde{\psi}_m \qquad (7)$$

Multiplying eq.[6] by ψ_m and eq.[7] by $\tilde{\psi}_m$ and subtracting the products gives:

$$(\tilde{k}_m^2 - k_m^2)\cdot\int_0^\infty \psi_m\tilde{\psi}_m\, dz = \int_0^\infty (\tilde{k}^2 - k_0^2)\psi_m\tilde{\psi}_m\, dz \qquad (8)$$

We take the following approximations:

$$\tilde{k}^2 - k_0^2 \approx \omega_0^2\left(\frac{-2\Delta c}{c_0^3}\right) \qquad (9)$$

$$(\tilde{k}_m + k_m) \approx 2k_m \qquad (10)$$

Then, we get

$$(\tilde{k}_m - k_m) = \frac{-\omega_0^2 \int_0^\infty \tilde{\psi}_m\psi_m\left(\frac{\Delta c}{c_0^3}\right)dz}{k_m \int_0^\infty \tilde{\psi}_m\psi_m\, dz} \qquad (11)$$

Substituting eq.(11) into eq.(5), the modal travel time perturbation is expressed in a "differential" form:

$$\delta t_m = \int_0^R dr\left[\frac{\partial}{\partial\omega}\left(\frac{-\omega_0^2\int_0^\infty \psi_m\tilde{\psi}_m\left(\frac{\Delta c}{c_0^3}\right)dz}{k_m\int \psi_m\tilde{\psi}_m\, dz}\right)\right] \qquad (12)$$

3. THE NONLINEARITY ANALYSIS

As we can see, that the nonlinearity is mainly caused by the deformation of eigenfunction in terms of $\tilde{\psi}_m$. By writting

$$\tilde{\psi}_m = \psi_m + \delta\psi_m \qquad (13)$$

Then, the linear and nonlinear constituent are decomposed as follows:

$$\delta t_m = \delta t_m^{(L)} + \delta t_m^{(NL)} \qquad (14)$$

where $\delta t_m^{(L)}$ is the linear constituent:

$$\delta t_m^{(L)} = \int_0^R dr\Big[\frac{\partial}{\partial\omega}\Big(\frac{-\omega^2}{k_m}\int_0^\infty \psi_m^2 \Big(\frac{\Delta c}{C_0^3}\Big)dz\Big)\Big] \qquad (15)$$

and $\delta t_m^{(NL)}$ is the nonlinear constituent:

$$\delta t_m^{(NL)} = \int_0^R dr\Big\{\frac{\partial}{\partial\omega}\Big[\frac{-\omega^2}{k_m}\Big(1+\int_0^\infty\psi_m\delta\psi_m dz\Big)^{-1} \times \qquad (16)$$

$$\times\Big(\int_0^\infty \psi_m\delta\psi_m\Big(\frac{\Delta c}{C_0^3}\Big)dz - \int_0^\infty \psi_m^2\Big(\frac{\Delta c}{C_0^3}\Big)\int\psi_m\delta\psi_m dz\Big)\Big]\Big\}$$

Obviously, the nonlinear term δt_m^{NL} vanishes when the deformation of eigenfunction $\delta\psi_m$ is not significant. However, there are some cases that the nonlinearity is significant and for these cases the conventional linear inversion scheme will not be appropriate. Numerical examples are presented in the following section.

4. NUMERICAL EXAMPLES

The ocean model for numerical computation consists of a canonical Munk profile [2] as background and a Gaussian eddy perturbation as follows:

$$C_0(z) = 1500\{1 + 0.0057[e^{-\eta} - (1-\eta)]\} \qquad (17)$$

$$\Delta c(r,z) = (DC)\cdot\exp\Big\{-\Big[\frac{r-r_E}{DR}\Big]^2 - \Big[\frac{z-z_E}{DZ}\Big]^2\Big\} \qquad (18)$$

where $\eta = 2(z-1000)/1000$. The parameters of a weak cold eddy and a strong warm eddy are: DR=100 Km, DZ=500 m, ZE=1000 m, DC=-6 (cold), and DC=+15 (warm) respectively. Modal travel time perturbations are calculated for acoustic frequency f=10 Hz and mode number m=1,2,3. The results for weak cold eddy are listed on Tab.1 and the results for strong warm eddy are listed on Tab.2.

Tab.1 Modal travel time perturbation for a weak cold eddy.

m	δt_m	δt_m	$\delta t_m^{(L)}$	$\Delta=\delta t_m^{(L)}-\delta t_m$	$E= \Delta/\delta t_m$ (%)
	eq.(3)	(12)	(15)	(ms)	
1	450.1	448.8	440.5	-9.6	2.1
2	369.4	369.1	341.6	-27.8	-7.5
3	257.5	257.5	245.3	-12.2	-4.7

Tab.2 Modal travel time perturbation for a strong warm eddy.

	δt_m	δt_m	$\delta t_m^{(L)}$	$\Delta=\delta t_m^{(L)}-\delta t_m$	$E= \Delta/\delta t_m$ (%)
m	eq.(3)	(12)	(15)	(ms)	
1	-806.3	-810.3	-1101.1	-294.7	36.5
2	-804.8	-811.0	- 853.9	- 49.1	6.1
3	-581.7	-586.0	- 613.2	- 31.5	5.4

CONCLUSIONS

(1).As we can see from Tab.1 and Tab.2 that the modal travel time perturbation calculated by differential formula eq.(12) is very close to the result ginen by the exact formula eq.(3).

(2).The nonlinearity of modal travel time perturbation caused by a weak cold eddy (DC=-6 m/s), as illustrated in Tab.1, is not significant. However, for a warm strong eddy (DC=+15 m/s), as illustrated in Tab.2, is significant.

Acknowledgement: This work was supported by NOAA and ONR.

REFERENCES

[1].D.Chapman and D.Ellies,"The group velocity of normal mode," J. Acoust. Soc. Am. 73, 973-979, 1983.

[2].W.Munk,"Sound channel in an exponentially stratified ocean with application of SOFAR," J. Acoust. Soc.Am.,55,220,1974.

ALGORITHMS FOR MAXIMIZING MATCHED FIELD PROCESSING OUTPUT USED IN A NEW APPROACH TO OCEAN ACOUSTIC TOMOGRAPHY

A. Tolstoy
Acoustics Division
Naval Research Lab
Wash DC 20375 USA

L.N. Frazer
Hawaii Institute of Geophysics
University of Hawaii
Honolulu, HI 96822 USA

Abstract

A new approach to ocean acoustic tomography uses matched field processing for narrow band, low frequency sources distributed around the region perimeter and detected on widely distributed vertical arrays. A key component to the success of this new approach is an algorithm to compute "the global maximum" of the processor outputs over the very large set of candidate environments In this paper we discuss algorithms based on the "back propagation" algorithms used in medical tomography but modified to allow for non-uniform values along each source-receiver path and weighted according to the length of the path segment of interest. Computational results to date show that the algorithms can result in extremely accurate and efficient tomographic solutions.

I. INTRODUCTION

Ocean acoustic tomography is a technique involving the transmission of acoustic fields through an ocean region and subsequently inferring the 3-D sound-speed profiles of the region by examining those fields. Over the last decade ocean tomography experiments have shown that examination of the acoustic multipath arrivals interpreted in terms of ray theoretic models can be highly effective (Munk and Wunsch, '79; Behringer et al., '82; Cornuelle et al., '89). However, such an approach requires "high" frequency signals (above 100 Hz), and so results will be degraded by such factors as uncertainties in the source/receiver locations, internal waves and tides, rough surface scattering, etc. The measurement process itself can be extremely time-consuming requiring weeks at sea to navigate the perimeter and send signals through the region. In addition, the use of high frequencies which attenuate rapidly as a function of range limits the size of the regions which can be sampled. More generally, working with data in the time domain requires high time resolution receivers to distinguish arrivals and to detect changes in those arrivals which result from sound-speed variability. Our new technique examines interference patterns across vertical arrays of hydrophones for single frequency (not time domain) low frequency data (10 - 30 Hz) modeled by highly accurate normal mode methods. The sources will be explosive shots dropped from an airplane flying around the perimeter, and so experimental time will decrease from weeks to days. The new technique will effectively transfer the burden from intense oceanographic surveys to intense computer demands.

II. APPROACH

The essence of the new approach is to find the family of sound-speed profiles which maximizes the matched field processing (MFP) power[1] computed for each vertical array receiving signals from the known shot sources That is, signals received at the arrays are cross-correlated with modeled signals which have propagated through candidate environments, and we seek to maximize those correlations. If the problem is properly posed, i.e., if we impose sufficient constraints, then the maximum MFP power will occur only for the true environment.

The first stage of the process is to characterize the environment in as few parameters as possible. Oceanographers have developed a method for deriving efficient basis functions, known as empirical orthogonal functions (EOFs), from measured data (Davis, '76) Consequently, an ocean region might be very accurately described in terms of only 2 or 3 EOFs. The simulated "double eddy" environment and their associated (modified) EOFs used for the results in this paper are described in detail in Tolstoy et al., '91.

The next stage is to grid the ocean region into cells where each cell corresponds to one sound-speed profile, i.e., 2 or 3 EOF coefficients. We also need to consider the geometry of our problem: how many vertical arrays will we use and where will we deploy them; how many sources will we use and where will we drop them? For the results to be discussed here we will use four vertical arrays and 36 sources distributed as shown in Fig. 1. Each array will have 28 phones spaced at 37.5 m for processing 20 Hz signals. The first phone will be just below the surface of the water and thus span the upper 1000 m of water where all the sound-speed variability is found We assume that the sound-speed profiles at the source and array cells have been measured and their EOF coefficients are known. The complete values of the dominant EOF coefficient are shown in Fig. 2.

Finally, we need an algorithm to perform the inversion, i.e., to compute the unknown EOF coefficients which maximize the MFP power at the arrays for all the source signals.

III. THE ALGORITHM

First, we shall initialize our algorithm with a simple test environment. In particular, we will assume that we know the *range* of possible values for the EOF coefficients throughout our region and then use their mid-range values as a first estimate for all the unknown values (see Fig 3) We know that this estimate is not very good because the MFP power computed at each array for each source is very low. In Fig. 4 we see a plot of the MFP power P_{rs} for each receiver-source path for $r = 1$. The maxima should be about 28 (the number of phones).

Let $\beta_1(ij), \beta_2(ij)$ denote the true EOF coefficients for the ijth cell Consider the ijth cell, and iterate through all possible values of the EOF coefficient β_1. Let $P^*_{rs}(ij)$ denote the maximum power found for the path from source s to array r intersecting the ijth cell (all other coefficients along the path are fixed), i.e.,

$$P^*_{rs}(ij) = \max_{\beta_1} P_{rs}.$$

Let $\beta^*_{1.rs}(ij)$ be the corresponding coefficient, and $\Delta_{rs}(ij)$ be the length of the path through the cell. Then, define the new coefficient estimate by

$$\hat{\beta}_1(ij) = \frac{\sum_{rs} \beta^*_{1.rs}(ij)\Delta_{rs}(ij)}{\sum_{rs} \Delta_{rs}(ij)}.$$

Next, consider β_2 and repeat the procedure. Then, proceed to the next cell. When all cells have been processed (one sweep), repeat from the first cell (note that all the cells may have changed their coefficients and so path contributions from the non ijth cell will have changed). For the results presented here, the process was stopped when the total power $P_{total} = \sum_{rs} P_{rs}$ was no longer increasing.

For the example discussed, we obtained excellent results for 31 sweeps with a maximum sound-speed error everywhere of less than 0.2 m/sec. However, there were other array configurations for which the algorithm stalled, i.e., stopped before giving good results. So, we also considered variations of the algorithm by selecting the $\hat{\beta}_1(ij)$ which simply maximized $\sum_{rs} P_{rs}\Delta_{rs}$, $\sum_{rs} \sqrt{P_{rs}}\Delta_{rs}$, or $\sum_{rs} P_{rs}\Delta^2_{rs}$. In general, we found that these variations sometimes improved results over the original but sometimes did not, and were also prone to stalling.

CONCLUSIONS

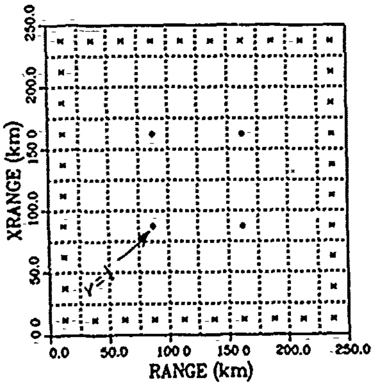
We conclude that efficient characterization of the environment, i.e., through the use of (modified) empirical orthogonal functions, plus careful source/array geometry, can result in highly accurate estimates of the 3-D sound-speed environment. In particular, we saw that 4 vertical arrays spanning the upper 1000 m of water and placed in the interior of the region of interest with shots distributed every 25 km along the perimeter resulted in maximum errors less than 0.2 m/sec for our 250 km per side square region and for the frequency 20 Hz. However, the algorithm and variations on it developed for the inversion can stall. We are presently working to find a remedy for this difficulty.

Bibliography

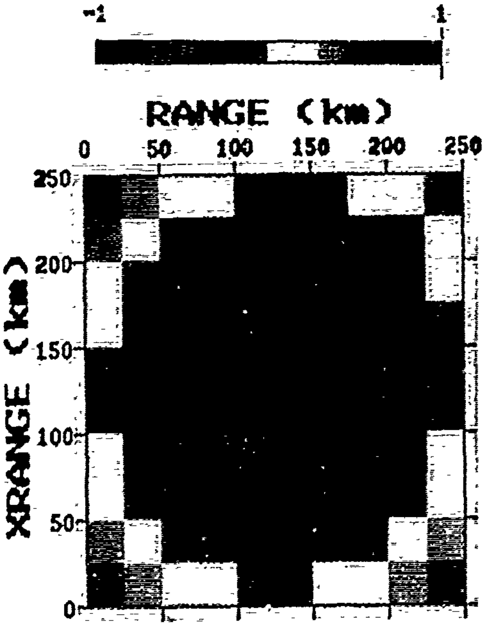
- Behringer, D., T. Birdsall, M. Brown, B. Cornuelle, R. Heinmiller, R. Knox, K. Metzer, W. Munk, J. Spiesberger, R. Spindel, D. Webb, P. Worcester, and C. Wunsch, "A demonstration of ocean acoustic tomography", Nature 200, 121-125 (1982).

[1] See Bucker (1976) and Fizell (1987) for details about MFP

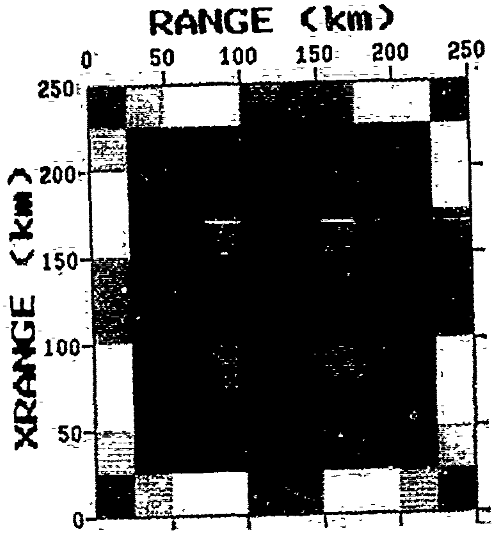
- H.P. Bucker, "Use of calculated sound fields and matched-field detection to locate sound sources in shallow water", J. Acoust. Soc. Am. 59, 368-373 (1976).
- Cornuelle, B., W. Munk, and P. Worcester, "Ocean acoustic tomography from ships", J. Geophys. Res. 94, 6232-6250 (1989).
- Davis, R.E., "Predictability of sea surface temperature and sea level pressure anomalies over the North Pacific Ocean", J. Phys. Ocean. 6, 249-266 (1976).
- Fizell, R.G., "Application of high-resolution processing to range and depth estimation using ambiguity function methods", J. Acoust. Soc. Am. 82, 606-613 (1987).
- Munk, W.H., and C. Wunsch, "Ocean acoustic tomography: a scheme for large scale monitoring", Deep Sea Res. 26A, 123-161 (1979).
- Tolstoy, A., O. Diachok, and L.N. Frazer, "Acoustic tomography via matched field processing", J. Acoust. Soc. Am. 89, (1991).



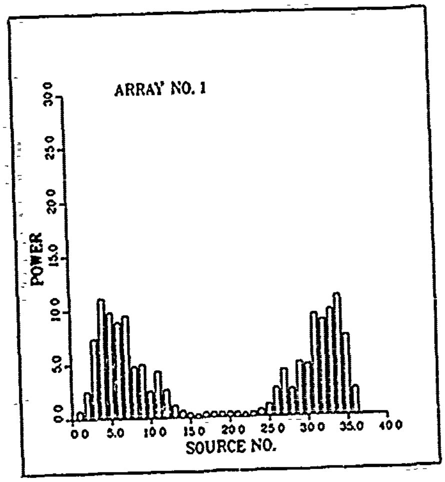
1. Distribution of 4 arrays (indicated by •) and 36 sources at 100 m depth (indicated by *) for 10 by 10 grid covering 250 by 250 square km.



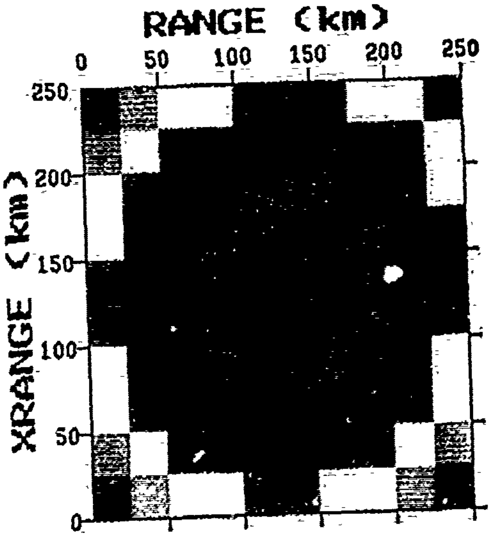
2. Plot of the dominant EOF coefficient $\beta_1(ij)$ as a function of range and cross-range. The scale has been normalised so that negative values range from -1 to 0, positive values from 0 to +1.



3. Initial estimates for EOF coefficient $\beta_1(ij)$ as a function of range and cross-range. The coefficients around the perimeter (where the sources are located) and at the arrays are known; otherwise, $\hat{\beta}_1$ = constant.



4. Initial estimates for MFP power P_{rs} at array $r = 1$ for each source s given initial environment of Fig. 3.



5. Final estimates for EOF coefficient $\beta_1(ij)$ as a function of range and cross-range. Compare to Fig. 2.

A NUMERICAL INVESTIGATION OF CHEBYSHEV SPECTRAL ELEMENT METHOD FOR ACOUSTIC WAVE PROPAGATION *

ENRICO PRIOLO
Osservatorio Geofisico Sperimentale
P. O. Box 2011
34016 Trieste, ITALY.

AND

GEZA SERIANI
Osservatorio Geofisico Sperimentale
P. O. Box 2011
34016 Trieste, ITALY.

Abstract - Seismic forward modelling is one of the foremost tools for investigating wave propagation in complex geological structures. Moreover, due to the complexity, both lithological and stratigraphical, that can be found in such structures, the use of a numerical method with great accuracy and flexibility is needed for correct results. In this respect, the Spectral Element Method (SPEM) which combines the accuracy of spectral techniques and the flexibility of finite element methods is well suited. The method has been applied to solve the acoustic wave equation. Accuracy and convergence properties of Chebyshev SPEM are discussed in the present work.

I. INTRODUCTION

The most widespread of the discrete numerical methods for modelling seismic wave propagation are the finite difference (FDM) [1],[6], the Fourier or pseudo spectral (FSM) [4],[7],[8], and the finite element (FEM) methods [9],[10]. Even though based on different mathematical approaches, all three methods rely on the space discretization of the geological structure to be modelled. In particular, the pseudo spectral method can be seen as a limiting case of the finite difference methods of increasing order and accuracy [3]. The main advantage of the FSM is its great accuracy, allowing for a lower number of grid points per minimum wavelength propagating in the model; a saving up to several orders of magnitude in computer memory and time can be realized. On the other hand, the FEM is well known for its flexibility in describing problems with complex geometries; irregular surfaces between different media can be defined with great accuracy. A method which combines FSM accuracy with FEM flexibility is desiderable for seismic wave modelling. A high-order finite element spectral method [12],[13], seems a good candidate for this.

In the present work, we shortly present the Chebyshev spectral element method and then we discuss the implementation and the numerical results obtained for the one dimensional acoustic wave equation. The accuracy and convergence properties of SPEM are investigated by comparison with standard finite element method and with an analytical solution. Estimations are carried out by computing a frequency error index function, that relates numerical and analytical results in the frequency domain, for interpolants of increasing order.

II. CHEBYSHEV SPECTRAL ELEMENT METHOD

The Fourier spectral technique is a particular case of the more general spectral methods (SPM) [5], and both spectral and finite element methods can be seen as particular cases of the class of discrete numerical techniques for solving differential equations known as the method of weighted residuals (MWR). With the MWR, the solution is obtained by minimizing the residual i.e., the error in the differential equation produced by using a truncated expansion instead of the exact solution, with respect to a suitable norm. To this end, a set of trial functions and a set of weight functions must be defined. The trial functions are used as the basis functions for the truncated series expansion of the solution; the weight functions are used to ensure that the differential equation is satisfied as closely as possible by the truncated series expansion. The choice of trial functions is different for the two methods. In the case of the spectral methods, the trial functions are infinitely differentiable global functions, while for the finite element methods, the domain is divided into small elements, and a trial function is specified in each element. In the latter case, the trial functions are with local support, and so well suited for handling complex geometries. Following the Galerkin approach, the weight functions are the same as the trial functions and are, therefore, smooth functions which individually satisfy the boundary conditions.

The Chebyshev SPEM that we present in this part of the paper is based on the idea of decomposing the spatial domain, where the physical problem must be solved, into subdomains, as in FEM, and then on each subdomain expressing the solution of the problem we are looking for by a truncated expansion of orthogonal polynomials, as in SPM. More specifically, in the case of one dimensional problems, we decompose the original spatial domain $\overline{\Omega}$ into non overlapping elements Ω_e , where $e = 1, \dots, n_e$, and n_e is the total number of elements. As approximating functions on each element Ω_e , we chose functions belonging to $\mathbb{P}_{N_e}$ space i.e., polynomials of degree $\leq N_e$ in x. Then the global approximating function is build up as a sum of the elemental approximating functions and, therefore, is a continuous function which is a piecewise polynomial defined on the decomposition $\tilde{\Omega}$ of the original domain Ω. The continuity of the derivative at the element interfaces is not satisfied for fixed N_e, but only as a consequence of the convergence process i.e., when all N_e tend to infinity.

We now discuss the construction of such

* This work was supported in part by the Commission of the European Communities under project EOS-1 (Exploration Oriented Seismic Modelling and Inversion), Contract N. JOUF-0033, as part of the GEOSCIENCE project within the framework of the JOULE R & D Programme (Section 3.1.1.b).

approximating functions using Chebyshev orthogonal polynomials. A function f(x), defined on the interval [-1, 1], can be approximated by a truncated expansion of Chebyshev polynomials as follows:

$$f(x) \sim I_N f(x) = \tilde{f}(x) = \sum_{k=0}^{N} \hat{c}_k T_k(x) \quad , \tag{1}$$

where T_k are the Chebyshev polynomials defined as

$$T_k(\cos \theta) = \cos k\theta \tag{2}$$

or, equivalently, with the recurrence relation

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x) \qquad \forall k \geq 1 \quad , \tag{3}$$

and $T_0(x) = 1$, $T_1(x) = x$. Using the orthogonality property of T_k and the Gauss-Lobatto quadrature formula, the expansion coefficients $\hat{c}_k$ are easily computed. It follows that the interpolant of f can be written as

$$I_N f(x) = \sum_{j=0}^{N} f(x_j)\varphi_j(x) \quad , \tag{4}$$

where $\varphi_j(x) \in \mathbb{P}_N$ are Lagrangian interpolants satisfying the relation $\varphi_j(x_k) = \delta_{jk}$ within the interval [-1, 1] and identically zero outside. The Lagrangian interpolants are given by

$$\varphi_j(x) = \frac{2}{N}\sum_{k=0}^{N} \frac{1}{\bar{c}_j \bar{c}_k} T_k(x_j)T_k(x) \quad , \tag{5}$$

$$\text{with} \quad \bar{c}_j = \begin{cases} 1 & \text{for} \quad j \neq 0, N \\ 2 & \text{for} \quad j = 0, N \end{cases} \quad , \tag{6}$$

and where x_j are the Chebyshev Gauss-Lobatto quadrature points

$$x_j = \cos\left(\frac{\pi j}{N}\right) \qquad \text{for} \quad j = 0, \ldots, N \quad . \tag{7}$$

In order to apply these interpolants to the spatial decomposition $\tilde{\Omega}$, we define the mapping $F^{(e)}(x): x \in \Omega_e \to \xi^{(e)} \in [-1, 1]$, between the points $x \in [a_e, a_{e+1}]$ of the element Ω_e and the local element coordinate system, by

$$\xi^{(e)} \equiv F^{(e)}(x) = \frac{2}{\Delta_e}(x - a_e) - 1 \quad , \tag{8}$$

with $\Delta_e = a_{e+1} - a_e$. The Chebyshev Gauss-Lobatto interpolants in the Ω_e element are then written as

$$\varphi_j^{(e)}(\xi^{(e)}) = \frac{2}{N_e}\sum_{k=0}^{N_e} \frac{1}{\bar{c}_j \bar{c}_k} T_k(\xi_j^{(e)})T_k(\xi^{(e)}) \quad , \tag{9}$$

where $\xi_j^{(e)}$ are the Gauss-Lobatto points in the local coordinate system.

For the spatial discretization of a two-dimensional problem, the Cartesian products of the Chebyshev-Lobatto points are used on each rectangular element Ω_e and the Lagrangian interpolants are represented using tensor products of Chebyshev polynomials.

III. ONE DIMENSIONAL WAVE EQUATION

To illustrate SPEM in practice, we use as model problem the one-dimensional wave equation which describe the propagation of longitudinal waves in an elastic rod. The initial boundary value problem for a rod of length L and with fixed boundaries can be stated as follows:

given u_0 and $\dot{u}_0$, find a continous function $u: \bar{\Omega} \times [0, T] \to \mathbb{R}$ such that it satisfies the equation

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0 \qquad \text{on} \quad \Omega \times (0, T) \quad , \tag{10}$$

$$\text{with} \quad u(0, t) = u(L, t) = 0 \qquad \forall t \in (0, T) \quad ,$$
$$\text{and} \quad u(x, 0) = u_0(x) \quad , \quad \dot{u}(x, 0) = \dot{u}_0(x) \qquad \forall x \in \Omega \quad ,$$

where the dot in $\dot{u}$ indicates partial differentiation with respect to time and where u(x, t) is the axial displacement, $c^2 = E/\rho$ is the characteristic velocity (velocity of sound), E is Young's modulus, ρ is the rod density, and $\bar{\Omega} = [0, L]$.

If we look for sufficiently regular solutions u, an equivalent, variational formulation of equation (10) is to find u(x, t) solution of

$$\frac{d^2}{dt^2}\int_{\Omega} w(x)\frac{1}{c^2}u(x, t)dx + \int_{\Omega} \frac{\partial w(x)}{\partial x} \frac{\partial u(x, t)}{\partial x}dx = 0 \quad , \tag{11}$$

for all functions w(x) which vanish on the boundaries and which, together with their first derivatives, are square integrable over Ω. In order to obtain the spectral-element approximation of the equation (9), we decompose $\bar{\Omega}$ into non-overlapping elements Ω_e , and on the decomposition $\tilde{\Omega}$ we define the following finite-dimensional spaces for the trial functions $\tilde{u}(x, t)$:

$$S_N = \left\{\tilde{u} \in C^0(\bar{\Omega} \times [0, T]) \mid \tilde{u}_e \in \mathbb{P}_{N_e};\ \tilde{u}(0, t) = \tilde{u}(L, t) = 0\right\},$$

and for the weight functions $\tilde{w}(x)$:

$$V_N = \left\{\tilde{w} \in C^0(\bar{\Omega}) \mid \tilde{w}_e \in \mathbb{P}_{N_e};\ \tilde{w}(0) = \tilde{w}(L) = 0\right\} \quad ,$$

where $\tilde{u}_e$ and $\tilde{w}_e$ denote the restriction to Ω_e of $\tilde{u}$ and $\tilde{w}$, respectively, and N denotes $\{N_1, \ldots, N_e\}$. Using previous definitions (9) for the interpolants on Ω_e, and according to the Galerkin approach, in the local coordinate system, functions $\tilde{u}_e$ and $\tilde{w}_e$ take the following form:

$$\tilde{u}_e(\xi^{(e)}, t) = \sum_{j=0}^{N_e} \tilde{u}_j^{(e)}(t)\ \varphi_j^{(e)}(\xi^{(e)}) \quad , \tag{12}$$

$$\tilde{w}_e(\xi^{(e)}) = \sum_{j=0}^{N_e} \tilde{w}_j^{(e)}\varphi_j^{(e)}(\xi^{(e)}) \quad , \tag{13}$$

where $\tilde{u}_j^{(e)}(t) = \tilde{u}_e(x_j^{(e)}, t)$ and $\tilde{w}_j^{(e)} = \tilde{w}_e(x_j^{(e)})$ are the grid values of the unknown approximate solution and of the weight functions, respectively. Using these approximating function spaces and the mapping $F^{(e)}(x)$ to solve equation (9), by straightforward computation, it can be shown that the one dimensional wave propagation problem becomes as follows:

given u_{0e} and $\dot{u}_{0e}$, find $\tilde{u} \in S_N$ such that for all $\tilde{w} \in V_N$ the following equations are satisfied:

$$\sum_{e=1}^{n_e}\left[\frac{d^2}{dt^2}(\tilde{w}_e, \frac{1}{c^2}\tilde{u}_e)_N + a(\tilde{w}_e, \tilde{u}_e)_N\right] = 0 \quad , \tag{14}$$

with $\tilde{u}_e(x,0) = \tilde{u}_{0e}(x)$, $\dot{\tilde{u}}_e(x,0) = \dot{\tilde{u}}_{0e}(x)$ $\forall e$,

where $a(\cdot\,,\cdot)_N$ and $(\cdot\,,\cdot)_N$ are symmetric, bilinear forms given by

$$(\tilde{w}_e, \frac{1}{c^2}\tilde{u}_e)_N = \sum_{i=0}^{N_e}\sum_{j=0}^{N_e} m_{ij}^{(e)}\ \tilde{w}_i^{(e)}\tilde{u}_j^{(e)}(t) \quad , \tag{15}$$

$$a(\tilde{w}_e, \tilde{u}_e)_N = \sum_{i=0}^{N_e}\sum_{j=0}^{N_e} k_{ij}^{(e)}\ \tilde{w}_i^{(e)}\tilde{u}_j^{(e)}(t) \quad , \tag{16}$$

and $m_{ij}^{(e)}$ and $k_{ij}^{(e)}$ are, respectively, the elemental mass and stiffness matrices

$$m_{ij}^{(e)} = \frac{2\Delta_e}{c^2(N_e)^2}\ \overline{a}_{ij} \sum_{p,q=0}^{N_e} \overline{a}_{pq}T_p(\xi_i^{(e)})T_q(\xi_j^{(e)})\ \overline{m}_{pq} \ , \tag{17}$$

$$k_{ij}^{(e)} = \frac{8}{\Delta_e(N_e)^2}\ \overline{a}_{ij} \sum_{p,q=0}^{N_e} \overline{a}_{pq}T_p(\xi_i^{(e)})T_q(\xi_j^{(e)})\ \overline{k}_{pq} \ , \tag{18}$$

with $\overline{a}_{ij} = 1/(\overline{c}_i\overline{c}_j)$. Here

$$\overline{m}_{pq} = \int_{-1}^{+1} T_pT_q dx = \begin{cases} 0 & \text{for } p+q \text{ odd} \\ \dfrac{1}{1-(p+q)^2} + \dfrac{1}{1-(p-q)^2} & \text{for } p+q \text{ even} \ , \end{cases} \tag{19}$$

and

$$\overline{k}_{pq} = \int_{-1}^{+1} \frac{dT_p}{dx}\frac{dT_q}{dx}dx = \begin{cases} 0 & \text{for } p+q \text{ odd} \\ \dfrac{pq}{2}[J_{|(p-q)/2|} - J_{|(p+q)/2|}] & \text{for } p+q \text{ even} \ , \end{cases} \tag{20}$$

where

$$J_n = \begin{cases} 0 & \text{for } n=0 \\ -4\displaystyle\sum_{r=1}^{n}\frac{1}{2r-1} & \text{for } n \geq 1 \ . \end{cases} \tag{21}$$

Let us define the connectivity matrix $B^{(e)}$ [11] as the matrix that topologically connects the approximate solution values $\tilde{u}_j^{(e)}$, in the local coordinate and numbering system, to the $\tilde{u}_s$ values, in the global coordinate and numbering system, such that

$$\tilde{u}_j^{(e)} = \sum_{s=1}^{n_g} B_{js}^{(e)}\ \tilde{u}_s \quad , \tag{22}$$

where n_g is the total number of nodes of the decomposition of the domain $\overline{\Omega}$, and $B^{(e)}$ is a Boolean matrix. Substituting expression (22) into (15-16) leads to

$$(\tilde{w}_e, \frac{1}{c^2}\tilde{u}_e)_N = \sum_{r=1}^{n_g}\sum_{s=1}^{n_g} M_{rs}^{(e)}\ \tilde{w}_r\tilde{u}_s(t) \quad , \tag{23}$$

$$a(\tilde{w}_e, \tilde{u}_e)_N = \sum_{r=1}^{n_g}\sum_{s=1}^{n_g} K_{rs}^{(e)}\ \tilde{w}_r\tilde{u}_s(t) \quad , \tag{24}$$

with $M^{(e)} = (B^{(e)})^T\ m^{(e)}\ B^{(e)}$, $K^{(e)} = (B^{(e)})^T\ k^{(e)}\ B^{(e)}$,

where $(B^{(e)})^T$ is the transpose matrix of $B^{(e)}$. Applying the relations (23-24) to the variational equation (14) and requiring that it be satisfied for all $\tilde{w}_r$, the spectral element approximation of our original equation finally leads us to solve

$$\sum_{s=1}^{n_g} M_{rs}\ \frac{d^2\tilde{u}_s(t)}{dt^2} + \sum_{s=1}^{n_g} K_{rs}\ \tilde{u}_s(t) = 0 \quad , \tag{25}$$

$$\text{with} \quad M = \sum_{e=1}^{n_e} M^{(e)} \quad , \quad K = \sum_{e=1}^{n_e} K^{(e)} \quad , \tag{26}$$

where $M = [M_{rs}]$ and $K = [M_{rs}]$ are, respectively, the mass and stiffness matrices obtained after a global nodal renumbering and assembly of the elemental matrices. The Dirichlet boundary conditions $\tilde{u}(0,t) = \tilde{u}(L,t) = 0$ are imposed by matrix condensation i.e., by eliminating the rows and columns corresponding to the two boundary points from the system. In the case of Neumann boundary conditions, they would have been taken into account naturally by the variational principle. Therefore, we have obtained an algebraic representation of the original problem which can be now stated as follows:

given the vectors U_0 and $\dot{U}_0$, find U such that it satisfies the equations

$$M\ \ddot{U} + K\ U = 0 \quad , \tag{27}$$

with $U(0) = U_0$, $\dot{U}(0) = \dot{U}_0$,

where the unknown vector U contains the values of the discrete solution $\tilde{u}$ at all Chebyshev points $x_j^{(e)}$, for $j = 0, \dots, N_e$ and for all $e = 0, \dots, n_e$ except for $x_0^{(1)} = 0$ and $x_N^{(n_e)} = L$. The matrices M and K are positive-definite, symmetric, and band-limited, the bandwidth being determined by the largest N_e. They can be easily computed for each chosen order of the interpolants.

To solve the system of linear, second order, ordinary differential equations with constant coefficients just derived, we must integrate over the time interval $[0,T]$. This is done by discretizing the time variable as $t_n = n\Delta t$, $0 \leq n \leq N_T$, where $\Delta t = T/N_T$, and N_T is the total number of time steps. At time t_n , the solution will be $U_n = U(t_n)$. From the different time integration schemes which are available, we used the Newmark central difference scheme which is an implicit two-step scheme, conditionally stable and second order accurate.

IV. ANALYSIS AND DISCUSSION OF SPEM

In order to investigate the accuracy of the method, comparisons were done between the analytical and numerical solutions of the one dimensional problem expressed by equation (10). The general solution of the problem is given by

$$u(x,t) = \frac{1}{2}\left[\hat{u}_0(x+ct) + \hat{u}_0(x-ct)\right] + \frac{1}{2c}\int_{x-ct}^{x+ct} \frac{d\hat{u}_0(s)}{dt}\,ds \quad , \tag{28}$$

where $\hat{u}_0$ and $d\hat{u}_0/dt$ are the odd 2L periodic extensions of u_0 and $\dot{u}_0$ to the entire real axis [2],[14].

Discrete numerical methods for solving a PDE introduce errors in the sought solution but they get progressively less as the mesh size becomes finer. Numerical modelling for wave propagation actually behaves as a low pass filter in the sense that low frequencies accurately propagate through the mesh whereas high frequencies are undesirably modified [10]. The most evident numerical effects (in the high-frequency band) are numerical attenuation, numerical anisotropy, dispersion in numerical phase and group velocity, and numerical polarization. Due to this fact, frequency analysis is a very suitable approach for the investigation of numerical modelling results. It makes it easier to determine the spectral band in which the equation is solved correctly; that is to find the minimum wavelength for which a given numerical method is accurate. Following this approach, we introduce the "frequency error-index", a complex function defined as

$$H(\omega) = \frac{U_{num}(\omega)}{U_{an}(\omega)} = |H(\omega)|\, e^{i\phi(\omega)} \quad , \tag{29}$$

where $U_{an}(\omega)$ and $U_{num}(\omega)$ are, respectively, the Fourier transforms of the analytical and numerical solutions that we want to compare. The frequency error-index makes the interpretation of the results effective since it is not affected by the spectrum of the chosen initial impulse but depends only on the numerical model. By considering both the real and imaginary part of the frequency error-index, the decay of accuracy due not only to amplitude variations (when $|H(\omega)| \neq 1$) but also to velocity dispersion (when $\phi = \mathrm{arctg}\left(\frac{\mathrm{Im}\{H(\omega)\}}{\mathrm{Re}\{H(\omega)\}}\right) \neq 0$) can be investigated.

Moreover, in order to get an idea of the rate of accuracy decay in time for each polynomial order, we can use the frequency error-index function. It is clearly important that, for large scale modelling in particular, not only efficiency (the number of grid-points per wavelength) but also accuracy of a numerical method be stable in time (after a certain reference distance of propagation).

We discuss now the numerical experiments performed and the results obtained. Let us denote the number of grid-points per wavelength by G. The following relations hold: $1/G = \Delta x/\lambda$, $0 < 1/G \leq .5$. As initial condition for the displacements, this function was chosen such that

$$u_0(x) = A\cos(2\pi\omega x)\exp[B(x-x_0)^2] \quad . \tag{30}$$

Analytical and numerical solutions were compared at travel distances in which the analytical solution reassumes the original form (because of the periodicity of the solution (28)). Results of simulations by FEM with polynomial orders N = 1, 2, 3, 4, and SPEM with orders N = 4, 8, 12, 15, 20, 30, 40, 60 have been collected. No orders less than N=4 were chosen for SPEM because of the fact that the Chebychev collocation-points for low polynomial order are very close to the equispaced points, so that no difference between the solutions obtained with the two methods can be expected in practice. During the experiments, the number of nodes was held constant ($n_g = 120$), and time steps were chosen for each polynomial order such as to ensure numerical stability.

The first set of experiments consisted in simulating the propagation for 2880 grid points of a broad band impulse through the elastic homogeneous rod, both with FEM and SPEM, but with different polynomial orders. By using a broad band impulse, we shall see that it is possible to identify, for each order of approximation, a low frequency band where the analytical and the numerical solutions agree. The maximum extreme of this interval gives the minimum G for which the solution is good. The $G_{min}(N)$ corresponding to the maximum of this band defines a wavelength $\lambda_{min}(N)$ $G_{min}(N)\Delta x$ which is the minimum wavelength that the model can propagate without appreciable errors.

In a second set of experiments, we used an impulse with a spectral band appropriately bounded in high frequency in order to prove that, in this case, the propagation errors are negligible.

As an example, figures 1 and 2 show the frequency error-index in both amplitude and phase, respectively, for SPEM with polynomial orde1 N=15 and for a propagation of 960 grid-points. The low-frequency band of the model, where analytical and numerical spectra are in good agreement, is well defined both by the amplitude and phase of the frequency error-index. The phase, however, shows very high instability just after the value of $G_{min} \approx 4$. From this curve it can be inferred that in the low-frequency band (that is for $\omega < 1/(G_{min}\Delta x)$) no phase shift exists. Similar results were obtained in the other cases.Thus no numerical dispersion is observed for waves which have a frequency in the correct range.

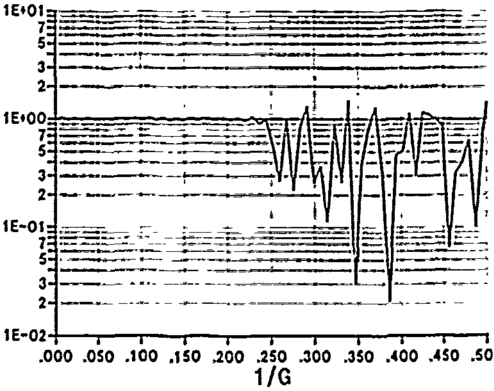


Fig. 1. Amplitude of the frequency error index for SPEM with polynomial order 15 at a distance of propagation of 960 grid-points.

Figure 3 contains a sketch of the values of $G_{min}(N)$ for the trials. For SPEM, since the inter nodal length is variable inside each element, two different estimates of G have been taken into account by using the mean value ("dx mean") and maximum value of dx ("dx max"). As the polynomial order increases, the values of

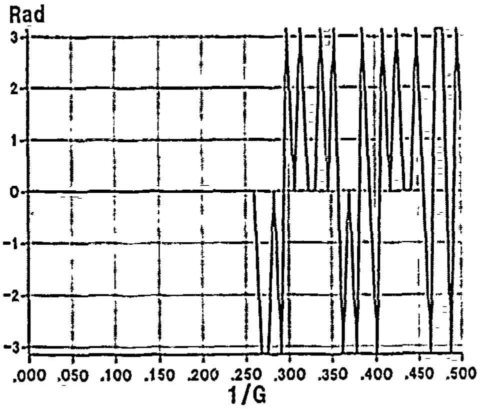


Fig. 2. Phase of the frequency error-index for SPEM with polynomial order 15 at a distance of propagation of 960 grid-points.

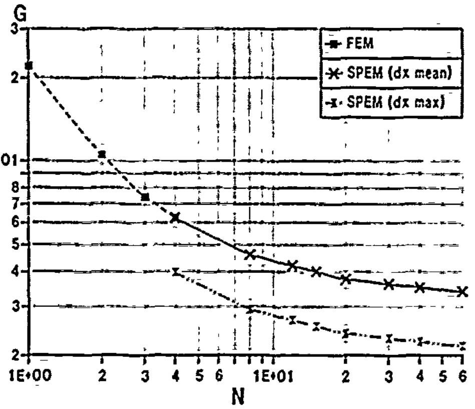


Fig. 3. Values for the minimum number of grid-points per wavelength (G) versus the polynomial order (N) for FEM and SPEM, at a distance of propagation of 960 grid-points. Estimations of G were done, taking into account the mean value of dx (dx mean) and the maximum of dx (dx max) respectively.

$G_{min}(N)$ for both estimates show a trend that looks asymptotic. Very low values of G_{min} (< 5) are reached for orders greater than 8; for N=60 a value of $G_{min}(60) = 3.4$ was found. The second type of estimate is interesting because, for a wave propagating in a discrete Chebyshev mesh, the sampling is minimum in the middle of the element where the inter-nodal length is maximum. It should be noted that according to the "dx mean" estimate, the asymptotic value is G≈3, but this corresponds to G≈2 according to the "dx max" estimate. That is, the theoretical limit of the spectral methods, G≈2, is reached locally with SPEM, which corresponds globally to G≈3 for the Chebyshev mesh.

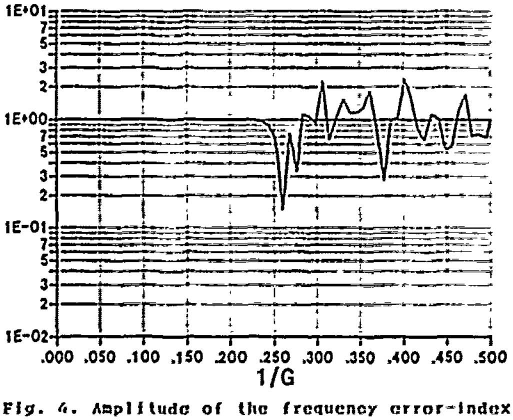


Fig. 4. Amplitude of the frequency error-index for the propagation of a low-frequency impulse for SPEM with polynomial order 15 at a distance of 960 grid-points.

Figures 4 shows the amplitude of the frequency error-index for the propagation of a pulse whose spectrum is inside the model frequency band (defined by $\omega < 1 / (G(N)\Delta x) = \omega_N$) for order N=15 (SPEM); for this order, $G_{min}(15) \approx 4$. For a propagation of 960 grid-points, the computed spectrum has in practice negligible errors, and the cut-off frequency is reached abruptly with almost no errors in the high part of the model spectrum band. The wave-forms at the same travel distance are shown in figures 5 and 6 using a broad-band impulse for a 3-order FEM and a 15-order SPEM, and in figure 7 using the band-limited impulse, respectively. In the last case, the errors are of the order of 10^{-4}. From a comparison, it is evident that SPEM is globally better performing and almost non-dispersive, as expected.

As a final point, G values were collected at different time steps and for different polynomial orders. In figure 8, curves are drawn representing the variation of G with travel distance for polynomial orders 1, 2, 3 (FEM), 4 (FEM and SPEM), and 15, 30 (SPEM). For low order FEM the curves show a quite high rate of increase, thus a large number of grid points per wavelength must be choosen for long

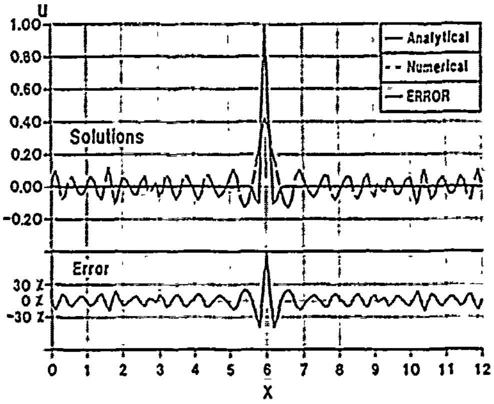


Fig. 5. Propagation of broad-band impulse for FEM with polynomial order 3 at a distance of 960 grid points. Wave forms of the analytical and numerical solutions and relative error are represented. The maximum error is 0.58.

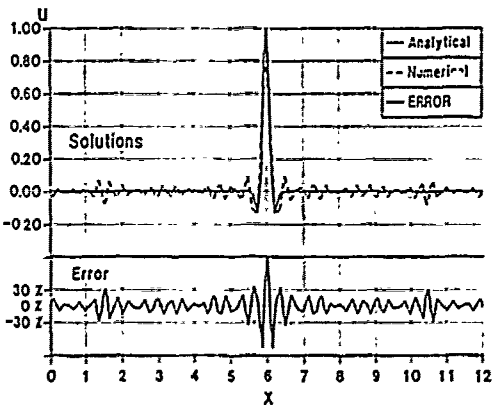


Fig. 6. As in figure 5, but for SPEM with polynomial order 15. The maximum error is 0.23.

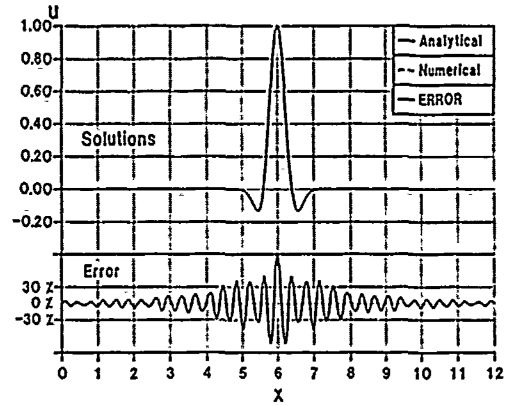


Fig. 7. Propagation of low frequency impulse for SPEM with polynomial order 15 at a distance of 960 grid-points. Wave-forms of the analytical and numerical solutions and relative error are represented. The maximum error is 0.00012.

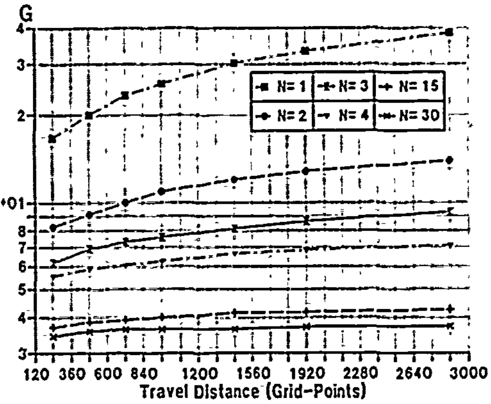


Fig. 8. Values for the minimum number of grid-points per wavelength (G) versus the distance (in grid-points) for polynomial order 1,2,3 (FEM), 4 (FEM/SPEM), and 15,30 (SPEM).

numerical simulations. For higher order SPEM, things are very different; the curves are slowly increasing, but they show very stable trends, even asymptotic. For N=15, values of G(15) from 3.7 to 4 may be detected; the maximum value for longer travel distances does not seem to exceed 4.2.

V. CONCLUSIONS

In this work the Chebyshev spectral element method has been presented. The SPEM solution of the one dimensional wave equation has been illustrated to demonstrate the feasibility of the method for wave equation problems. SPEM accuracy and convergence properties have been investigated numerically by computing the frequency error-index function for a large set of numerical experiments. As a conclusion, for the high-order Chebyshev spectral element method applied to the acoustic wave equation, the main results can be summarized as follows:

- low values of G, the number of grid points per wavelength, can be achieved (close to the theoretical minimum);
- great accuracy for the propagation of waves whose spectrum lies below the wavelength corresponding to the correct $G_{min}(N)$ is obtained;
- high-order schemes are very stable in time with respect to G and accuracy; they show almost no dispersion;
- the order of the interpolants can be changed very easily.

REFERENCES

[1] Boore, D.M., "Finite difference methods for seismic wave propagation in heterogeneous materials", In Methods in computational physics, vol.11, 1-37, ed. B.A. Bolt, Academic Press, New York, 1972.

[2] Duchateau, P., Zachmann, D.W., "Partial Differential equations", Shaum Outline Series in Mathematics, McGraw-Hill Book Comapany, 1986.

[3] Fornberg, B., "The Pseudospectral Method: Comparisons with Finite Differences for the Elastic Wave Equation", Geophysics, 52, 483-501,1987.

[4] Gazdag, J., "Modeling of the Acoustic Wave Equation with Transform Methods", Geophysics, 46, 854-859, 1981.

[5] Gottlieb, D., Orszag, S., "Numerical Analysis of Spectral Methods", CBMS-NSF series, SIAM, Philadelphia, 1977.

[6] Kelly, K.R., Ward, R.W., Treitel, S., and Alford, R.M., "Synthetic Seismograms: A Finite-Difference Approach", Geophysics, 41, 2-27, 1976.

[7] Kosloff, D., and Baysal, E., "Forward Modeling by Fourier Method", Geophysics, 47, 1402-1412, 1982.

[8] Kosloff, D., Reshef, M., and Loewenthal, D., "Elastic Wave Calculations by the Fourier Method", Bull. Seism. Soc. Am., 74, 875-899, 1984.

[9] Lysmer, J., and Drake, L. A., "A Finite element method for seismology", In Methods in computational physics, ed. B.A. Bolt, Academic Press, New York, vol.11, 181-216, 1972.

[10] Marfurt, K.J., "Accuracy of Finite-Difference and Finite-Element Modeling of the Scalar and Elastic Wave Equations", Geophysics, 49, 533-549, 1984.

[11] Oden, J.T., "Finite element applications in mathematical physics", In The mathematics of FEM and applications, ed. Whiteman, Academic press, New York, 239-282, 1973.

[12] Patera, A.T., "A spectral element method for fluid dynamics: laminar flow in a channel expansion", J. of Comput. Physics, 54, 468-488, 1984.

[13] Seriani, G., Priolo, E., "High Order Spectral Element Method (SPEM) for Acoustic Wave Propagation", 53rd EAEG Meeting Expanded Abstracts, Florence, Italy, May 1991.

[.4] Tyn Mynt-U, "Partial Differential Equations for Scientists and Engineers", III Ed., North Holland, 1987.

ACOUSTIC PROPAGATION IN OCEAN SEDIMENTS

by

Werner E. Kohler
Department of Mathematics
Virginia Tech
Blacksburg, Va 24061

George C. Papanicolaou
Courant Institute of Mathematical Sciences
New York, NY 10012

and

Benjamin White
Exxon Research & Engineering Company
Annandale, NJ 08801

Abstract - We consider CW radiation by a point source in the presence of a two-scale randomly-layered medium (a slab of finite thickness or a random half space). The source can be exterior to or buried within the random layering. Expressions are obtained for the first moment and two-point correlation function of the acoustic pressure.

Introduction.

Abyssal plains comprise much of the world's ocean bottom and consist of a sediment layer (nominally one kilometer thick) lying upon a rock basement. The sediment layer is itself of heterogeneous composition and possesses acoustic constitutive parameters (density and sound speed) that vary on two length scales [1]. On the one hand, the formative deposition processes have created a material, layered with clay, silt, pelagic remains and the like, whose constitutive parameters fluctuate rapidly with depth (on the scale of centimeters). On the other hand, compactification due to overbearing has imparted a slow scale or macroscopic variation to the mean value of these parameters; mean sediment sound speed, for example, increases with depth at a rate of roughly one sec^{-1} [2], and thus undergoes an $O(1)$ change on the scale of a kilometer. In the transverse directions, the acoustic parameters of the sediment layer are believed to remain constant to a degree that makes one-dimensional modeling a reasonable idealization. The propagation model, therefore, that proves useful in the study of acoustic propagation within the ocean sediments is that of a transversely homogeneous (slightly dissipative) slab whose acoustic constitutive parameters have a rapid fluctuation structure superposed upon a slow mean variation in the depth direction.

A small parameter ϵ ($0 < \epsilon << 1$) is used to characterize the scales. The correlation length of the fine scale constitutive parameter fluctuations is assumed to be $O(\epsilon^2)$ while the macroscopic mean variations are assumed to be $O(1)$. The wavelength of the CW source radiation is assumed to have an intermediate $O(\epsilon)$ spatial extent. This interpolating wavelength regime is the most interesting. The wavelength spans many correlation lengths and a useful limiting probabilistic description of the field quantities of interest is possible. Yet, wavelength is small relative to the macroscale and high frequency (WKB) approximations can be exploited. For the ocean sediment environment, with a macroscale of one kilometer and a correlation length of ten centimeters, ϵ^2 would equal 10^{-4} and the corresponding wavelength would therefore be ten meters (roughly 150 Hz.).

Results presented here comprise a small portion of a comprehensive theory, developed in collaboration with Mark Asch and Marie Postel. This theory, which encompasses pulse as well as CW excitation and inverse as well as direct problems, is summarized in [3].

The Problem.

Consider a randomly layered slab occupying $-L < z < 0$, with a CW point acoustic source at height z_s above the origin. Assuming an $e^{-i\omega t/\epsilon}$ time dependence, the (scaled and dimensionless) acoustic equations become:

$$\begin{aligned} \frac{-i\omega}{\epsilon}\rho\underline{u} + \nabla p &= \epsilon\delta(x)\delta(y)\delta(z-z_s)\underline{e} \\ \frac{-i\omega}{\epsilon}K^{-1}p + \nabla\cdot\underline{u} &= 0 \end{aligned} \tag{1}$$

where $\underline{e} = (e_1, e_2, e_3)$ is a constant unit vector. The bulk modulus K and density ρ vary with depth as follows:

$$\begin{aligned} K^{-1} &= \begin{cases} K_0^{-1}, & z > 0 \\ K_1^{-1}(z)[1 + \nu(z, z/\epsilon^2)], & -L < z < 0 \\ K_2^{-1}, & z < -L \end{cases} \\ \rho &= \begin{cases} \rho_0, & z > 0 \\ \rho_1(z)[1 + \eta(z, z/\epsilon^2)], & -L < z < 0 \\ \rho_2, & z < -L \end{cases} \end{aligned} \tag{2}$$

where η and ν are zero mean stochastic processes. Fourier transformation of the transverse spatial coordinates leads to the following system of stochastic ordinary differential equations.

$$\begin{aligned} \frac{d\hat{p}}{dz} &= \frac{i\omega}{\epsilon}\hat{u}_3 + \epsilon\delta(z-z_s)e_3 \\ \frac{d\hat{u}_3}{dz} &= \frac{i\omega}{\epsilon}(K^{-1} - \rho^{-1}\kappa^2)\hat{p} + \epsilon\rho^{-1}\delta(z-z_s)\underline{\kappa}\cdot\underline{e} \end{aligned} \tag{3}$$

where $\hat{u}_3$ is the z component of (transformed) particle velocity, $\underline{\kappa}$ is the scaled transverse slowness and $\kappa^2 = \underline{\kappa}\ \underline{\kappa}$. Continuity of pressure and normal particle velocity at interfaces $z = -L$ and $z = 0$ and outgoing radiation conditions as $z \to \pm\infty$ complete the problem specification.

Some Results.

Using the asymptotic theory derived in [3], we can characterize the first and second moments of the (reflected and transmitted) pressure For brevity, we here give results for the simplest configuration, for the pressure reflected from a lossless random half space (i e $L \to \infty$) having a constant deterministic background and only random sound speed fluctuations. For this case, the coherent reflected pressure is:

$$E\{p_{refl}(x,y,0,t)\} \sim \frac{-i\omega e^{\frac{-i\omega}{\epsilon}(t - c_0^{-1}R)}}{4\pi c_0 R}\left[\frac{xe_1 + ye_2 - z_s e_3}{R}\right]\cdot \left[\frac{\rho_1\rho_0^{-1}z_s[c_0^2c_1^{-2}R^2 - r^2]^{-1/2} - 1}{\rho_1\rho_0^{-1}z_s[c_0^2c_1^{-2}R^2 - r^2]^{-1/2} + 1}\right] \tag{4a}$$

where

$$r^2 \equiv x^2 + y^2, \quad R^2 \equiv r^2 + z_s^2, \quad c_j \equiv (K_j/\rho_j)^{1/2}, \ j = 0, 1. \tag{4b}$$

Thus the coherent reflected pressure is determined by reflection from an effective medium, characterized by ρ_1 and c_1. When the problem is further simplified by assuming that the medium is matched (i.e $c_1 = c_0$) and the source is placed on the interface $z = 0$, the reflected pressure intensity becomes:

$$E\{|p_{refl}(x,y,0,t)|^2\} \sim \frac{\omega^2}{32\pi^2 l r}\int_0^1 dy\, y^2\sqrt{1-y^2}\left[y\sqrt{1-y^2} + \frac{r}{2l}\right]^{-2} \tag{5a}$$

where the localization length l is given by

$$l = 2c_1^2 \left[\omega^2 \int_0^\infty E\{\nu(\sigma)\nu(0)\} d\sigma\right]^{-1} \tag{5b}$$

Note that when radial distance from the origin is much greater than a localization length, i.e. when $\frac{r}{2l} >> 1$, the reflected intensity becomes approximately equal to:

$$E\{|p_{refl}(x,y,0,t)|^2\} \approx \frac{l\omega^2}{128\pi c_1^2 r^3} \tag{6}$$

Thus, the theory predicts that a substantial amount of acoustic energy is reflected from the randomly layered half-space.

References.

[1] B. E. Tucholke, Acoustic Environment of the Hatteras and Nares Abyssal Plains, western North Atlantic Ocean, Determined from Velocities and Physical Properties of Sediment Cores, J. Acoust. Soc. Amer., 68, 1376 - 1390, 1980.

[2] E. Hamilton, Geoacoustic Models of the Sea Floor, in: L. Hampton, ed., Physics of Sound in Marine Sediments, Plenum, New York (1974), 181 - 221.

[3] M. Asch, W. Kohler, G. Papanicolaou, M. Postel and B. White, Frequency Content of Randomly Scattered Signals, to appear in SIAM Review, 1991.

ACTIVE TIME DELAY AND PHASIS ESTIMATION IN UNDERWATER ACOUSTICS

G. JOURDAIN

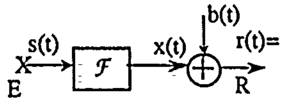
CEPHAG/IEG BP 46, 38402 St Martin d'Hères, France.

Abstract

We are interested in active identification of the multipath underwater (u.w) propagation, ie in the estimation of path parameters (delays, amplitudes and phases). This completes the prediction given by the wave propagation equation. The classical method for identifying the u.w channel response is presented, and some examples of u.w multipath channels are given. The problem of path parameter estimation becomes more difficult when the paths are very close. We have proposed some high resolution methods - or joint estimation methods - which enable to solve close paths, estimate their parameters and follow their time variations. Some results applied to u.w data are finally given.

I - MULTIPATH PROPAGATION IN U.W ACOUSTICS

It is now well admitted the u.w channel can be modelled as a linear filter $\mathcal{F}$ between an emitter E and a receiver R, moreover the additive noise.



$$r(t)=\mathcal{F}(s(t)) + b(t) \qquad (1)$$

Fig 1 : The u.w channel model

This filter takes account of the *energetic* aspect (absorption, diffraction...) but also of *temporal* and *frequential* distorsions, and particularly *multipath* effects.
According to the propagation and geophysical conditions (E, R, bottom...), to the carrier frequency ν_0, and also the *time scale* of interest, this filter can be modelled either random, or deterministic, or time varying [1] . Anayway the description of $\mathcal{F}$, and its time or stochastic variations, is a main complement to the prediction given by the solution of sound propagation equation in u.w acoustics (ray or mode theory, or hyperbolic equation...) which only gives an *approximate* and *static* solution.

As the emitted and received signals are always band pass around ν_0, we use the *complex amplitudes* relative to ν_0, denoted c(t) for emission, y(t) for reception, n(t) for noise, and H(t) the corresponding channel band pass impulse response (i.r). The *multipath* propagation is traduced by

$$y(t) = \sum_{k=1}^{p} \alpha_k e^{i\phi_k} \; c(t-\tau_k) + n(t) \qquad (2)$$

$$H(t) = \sum_{k=1}^{p} \alpha_k e^{i\phi_k} \; \delta(t-\tau_k) \qquad (3)$$

where the parameters α_k, ϕ_k, τ_k are assumed here constant, but they can be random, time varying...α_k traduces the energetic level transmitted over the delay τ_k, the phasis ϕ_k includes the phasis relative to the path k plus a delay term $(-2\pi\nu_0\tau_k)$ plus an eventual demodulation phasis.

II - ESTIMATION OF THE IR. EXPERIMENTAL EXAMPLES

The problem of interest in this paper is the estimation of the u.w channel. Let us note we perform here *active identification*, and not passive time delay estimation (see for ex [2]). It has been shown [3] the minimum output error solution for the channel identification is to *emit large WT product* (W bandwith, T duration) *signals*, and to *cross correlate* the channel output with a copy of emission In baseband notation this leads to

$$\Gamma_{yc}(t) = \sum_{k=1}^{p} \alpha_k e^{i\phi_k} \; \Gamma_c(t-\tau_k) + b'(t) \qquad (4)$$

where Γ_c is c(t) auto correlation function and $b'(t) = \Gamma_{yn}$.

Two examples are given below, coming from 2 sea experiments. The first (fig.2a) has been obtained in the following conditions : shallow water ; E/R distance = 4 km; $\nu_0 \sim$ 500 Hz.. The 2nd example is deep water, and more stable because ν_0 = 60 Hz , $d_{E/R}$ = 140 km.
The multipath structure is evident on both figures with different typical time variations of parameters.

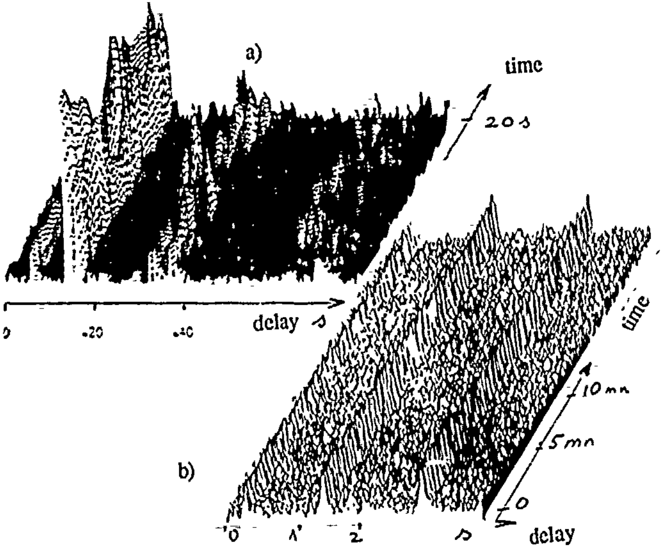


Fig. 2 : Examples of u.w responses H(τ)

From many sea experiments [1] one can consider the positions τ_k are always stable enough ; the amplitudes α_k are often rapidly varying (as soon as ν_0 > 100 Hz) , the phases ϕ_k are not seen here, they are always slowly varying, and very often the path *phases differences* $\phi_k - \phi_j$ are very stable.
For a precise estimation of parameters the problem is no longer to estimate H(t), but *the set of parameters* $\{\alpha_k, \phi_k, \tau_k\}$ of the model (3).

III - PARAMETER ESTIMATION OF THE MODEL (3)

III-1. One path case : In this case (p = 1), and if n(t) is white, gaussian, with psd γ_0, the above mentioned cross correlation (CCOR) leads to the Maximum Likelihood (M.L) estimates of τ,α,ϕ, denoted by $\hat{\tau}$, $\hat{\alpha}$ $\hat{\phi}$ · $\hat{\tau}$ is the delay for which this CCOR is maximum, $\hat{\alpha}$ and $\hat{\phi}$ are the modulus and phasis of the maximum of CCOR. It is well known these estimates are asymptotically unbiaised and their variance is bounded by Cramer Rao bound (the performance is directly connected to SNR and c(t) effective signal bandwith - see [3]).
As soon as p ≥ 2, the calculus of the structure of joint ML estimate of the parameter set becomes complex, and the performance calculus too, particularly when the paths are very close (closer than 1/W), ie when they are no longer distinguished by the CCOR processing. (Let us note the cases of 2 and 3 joint paths have been treated [4]). So in the examples of fig 2, some of the paths are well separated, whereas some other are undistinguishable.

III-2. Close paths estimation . Different solutions have been proposed in order to solve and estimate close paths *after* the CCOR step , this first step is important because SNR is improved and some paths are already solved. A first kind of methods (see [5] and references) consists in transposing the high resolution (HR) methods well known in spatial or frequencial filtering (Music methods...). We have used the Tufts Kumaresan method for the case of fig 2b [5]. Without detailing this method, let us say i) it needs a deconvolution step, ii) the estimation of the amplitudes $\alpha_k e^{i\phi_k}$ is uncoupled of τ_k estimation,which is not optimal.

In any case the number p of paths must be first estimated.
Let us see now a second kind of method which directly *joint estimates* the set of parameters of the model (3).

IV - JOINT ESTIMATION OF DELAYS, PHASES AND AMPLITUDES

IV 1 The method : By time sampling the equation (4), the following matrix/vector equation is obtained

$$\underline{\Gamma} = M(\underline{\tau}, \underline{\Gamma}c)\,\underline{a} + \underline{b} \qquad (5)$$

where M(.,.) is the *model matrix* (where $\underline{\tau}$ is unknown, but $\underline{\Gamma}c$ is known) and $\underline{a}$ is the unknown complex amplitude vector of $\alpha_k e i\phi_k$
The minimization of

$$J \triangleq \|\underline{\Gamma} - M\underline{a}\|^2 \qquad (6)$$

versus the whole $\underline{\tau}$ and $\underline{a}$ parameters is the optimal least square solution (or ML solution if $\underline{b}$ is gaussian) In this case too, the number p must be first estimated by a detection criterion for example ; practically one can also test several values of p. These optimal estimates are [4] :

$$\left.\begin{aligned} \hat{\tau} &= \text{Arg min} \| P^{\perp} y \|^2 \\ \hat{\alpha} &= \text{Arg min } J \end{aligned}\right\} \quad \text{where } P = I - M(M^t M^{-1})M^t \qquad (7)$$

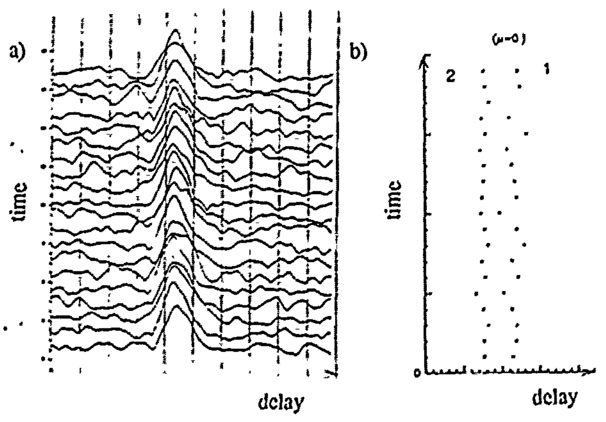
The minimization of J is performed by a gradient descent algorithm judiciously initialized [4].

IV 2 Sequential estimation : The above minimization of J is applied to each received data $\underline{\Gamma}$, ie each CCOR $\Gamma yc(t)$. In our u w case, as shown in fig 2, one always disposes of several successive $\Gamma yc(t)$. We have recently proposed [4] to improve the procedure by taking account of *successive* intercorrelations. The above non linear estimation method enables to easy introduce this as a priori information This is equivalent of adding a constraint equation on some parameter $\underline{\theta}$ in (6) ("regularization"). So now one tries to minimize for each record $\Gamma y c n$

$$J_n = \|\underline{\Gamma}_n - M(\underline{\tau}_n, \underline{\Gamma}_c)\,\underline{a}_n\|^2 + \mu\|\underline{\theta}_n - \underline{\theta}_{n-1}\|^2 \qquad (8)$$

In the u.w case, as the delay positions τ_k are stable enough from one record to another, we use only $\underline{\tau}$ as constraint parameter $\underline{\theta}$ in (8).

IV 3 Results : In the fig 2b, there are 3 "main paths" but there are perhaps some unsolve paths inside them and the ray tracing predicts 2 close paths inside the second "path". First a zoom is made of this second "path" (fig 3a). One tries to identify a model like (5) in it.



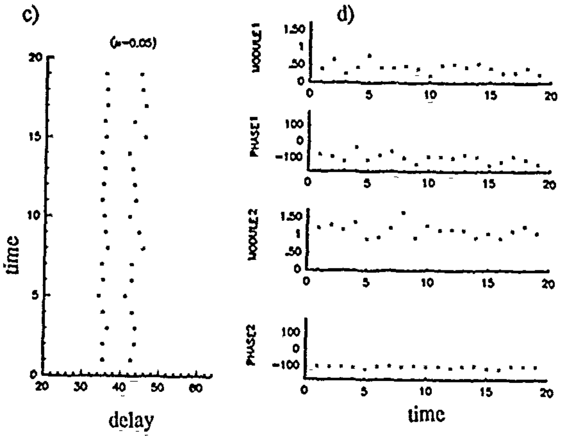


Fig 3 : Joint estimation of a couple of $\underline{\tau}$, $\underline{\alpha}$ and $\underline{\phi}$

The results are given in fig 3b . μ = 0 corresponds to the minimization of J (6) , and in fig 3c . μ = 0.05 corresponds to the minimization of Jn (8). Now two paths are well exhibited and identified, and their time variations are followed. Their modulus and phasis are given in fig 3d.

V - CONCLUSION

The principle and some results of the estimation of the u.w channel i.r have been given , particularly for the multipath case, the estimation of delays, amplitudes and phases has been studied. The CCOR step improves SNR and gives a first path separation. The improvement given by the second step (minimization of J and, better, Jn) enables to solve and characterize close paths. The method performs even when the number of triple parameters is not small (for ex.. 6 paths).

This estimation is important in two kinds of objectives :

i) the knowledge of the u.w propagation - and there is now a large interest in it, for ex. in the international u.w Acoustic Tomography Project [6]:), which is based on the precise time delays estimation.

ii) the elaboration of efficient detection or communication u.w systems : it is important to know the structure and variations of multipath to compensate them.

This work has been partly supported by French DCN, and also an Ifremer contract.

VI - REFERENCES

[1] G. Jourdain, Advanced methods for the investigation of the underwater channel - Underwater Acoustic Data Processing - Ed. by YT. CHAN, Nato ASI series, Vol. 161, 1989.

[2] AH. Quazi, An overview on the time delay estimate in active and passive systems for target localization - IEEE ASSP - 29 n° 3, pp 527-533, Juin 1981

[3] G. Jourdain, MA. Pallas, Multiple time delay estimation in u.w acoustic propagation. Stochastic Processes in u.w acoustics - CR Baker Ed Springer Verlag, 1986

[4] V. Nimier, Contribution à l'estimation des paramètres caractérisant la propagation par trajets multiples, Thèse de doctorat de l'INPG, 7 novembre 1990.

[5] M.A. Pallas, G. Jourdain, Active high resolution time delay estimation for large WT signals, IEEE ASSP 1991 (à paraître)

[6] RC. Spindel, Signal processing in ocean tomography, adaptative methods in u.w acoustic. HG Urbon, Ed Dordrecht/Boston Reidel 1985.

APPLICATION OF COMPUTATIONAL FLUID DYNAMICS IN HIGH SPEED AEROPROPULSION

Louis A. Povinelli*

Internal Fluid Mechanics Division
NASA Lewis Research Center
Cleveland, Ohio 44135 USA

Abstract - This paper describes the application of computational fluid dynamics to a hypersonic propulsion system, and serves as an introduction for this session. An overview of the problems associated with a propulsion system of this type is presented, highlighting the special role that CFD plays in the design.

I. INTRODUCTION

CFD has demonstrated some rather significant and spectacular achievements for aeronautical vehicles and their flow fields over the last 15 years. More recently some of these gains have been brought to bear on propulsion systems for aircraft; namely in the complex, wall bounded internal flows with energy addition inside of engines. An extensive activity has been pursued in attempting to validate numerical methods using data obtained from component (inlet, ducts, nozzles, combustors) testing. The computer codes developed have incorporated extensive physical and chemical modeling or closures, as well as utilizing multi-dimensions and sophisticated grid generation and adaptation methods. Due to the extremely complex nature of internal flow of engines, including rotating machinery, the validation and calibration of propulsion codes has proceeded slowly but steadily. With the resurgence of interest in a hypersonic air-breathing aircraft, i.e., the National Aerospace Plane, CFD efforts have been focused strongly on its proposed engine cycle. That cycle as envisioned currently relies on a supersonic combustion ramjet (Mach 6 to 15) used in conjunction with an accelerator up to Mach 6 and a rocket engine from approximately Mach 15 to orbital speed. The challenge to the scientific community is to develop accurate numerical simulations for this type of aircraft and propulsion system. Since scramjets have not been demonstrated on any propulsion system, the cycle must yet be proven feasible. In the absence of any flight data and the meager prospect of obtaining any data above flight Mach numbers of 8 in the near future, CFD becomes the tool of necessity for design of the engine and vehicle. It is worthwhile to point out, that the highly blended propulsion system makes it impossible to consider the engine without considering the influence of the airframe.

In this session, we shall concentrate on the hypersonic propulsion system and our progress is developing reliable CFD codes for analysis and design of the propulsion components. In particular, we shall look at the speed range corresponding to scramjet operation.

II. AIR CAPTURE

It should be noted at the outset that the air reaching the inlet face has experienced a rather trying time from the moment it traverses the shock wave at the aircraft nose. Depending on flight Mach number, the air may be dissociated and ionized as it moves along the underside of the aircraft. This air may undergo catalytic effects at the vehicle wall as well. Chemical and thermal nonequilibrium effects need to be modeled as well as catalicity. At the inlet plane, a substantial boundary layer has been developed on the ramp side of the inlet. It may be laminar or turbulent, or possibly transitional in nature. Shock waves from the cowl leading edge and the inlet sidewalls introduce additional complexities, such as shock-boundary layer interaction, which need to be modeled in the CFD codes. An example of such interactions is shown in a videotape. The computations, which are the result of the work of Benson and Reddy, at NASA Lewis Research Center (LeRC) illustrates a further trial for the captured airflow. The particle tracing shows that the intersection of the cowl and ramp shocks with the sidewall boundary layers causes a movement of the lower energy wall flow towards a narrow region. Cross-sectional inspection of the computed flow field reveals a vortex-like feature. At the throat, this flow behavior extends over a sufficient portion so as to cause concern regarding performance and stability of the inlet. Other rectangular inlets, such as the sidewall compression type, also experience similar effects. In this session we shall hear further discussion on inlets. Additional information is provided by this author in AGARD proceedings. Comparison of the computer inlet flow field and experimental data have shown good general aerodynamic agreement. However, in the regions where strong viscous effects are present, the agreement is marginal. Both transition and turbulence modeling improvements are required. Compressibility effects on turbulence modeling is currently being pursued as well as second moment closure by Shih and his

*Deputy Chief.

cohorts at the ICOMP Center for Modeling of Turbulence and Transition at LeRC. Comparisons of heat transfer data on the inlet walls with computed results show significant differences that need to be reconciled if CFD is going to affect scramjet thermal heating design.

III. MIXING AND BURNING

Mixing and combustion of hydrogen in supersonic flow (Mach 1.5 to 7) is the critical issue to be solved for the success of scramjets. A seminal contribution to supersonic mixing was put forward by this author regarding the generation of streamwise vorticity for mixing enhancement. Current generic engine combustors rely on the concept of vorticity generation. The method of generation differs only in detail from that of this author, but not in principle. It employs swept leading edges at angle of attack to the supersonic stream to promote vorticity. Shock vortex interaction was also proposed as a means of mixing enhancement, but it is less influential than vorticity generation. The basic issue revolves about the fact that jet penetration into supersonic flows is limited to about 10 jet diameters; an amount that is insufficient for a combustion chamber. Struts protruding into the stream produce the anticipated and predictable drag, must be cooled, and must be retracted over a portion of the flight range. Some current research centers on the vorticity generation concept mentioned above using swept wall injectors with fuel injection from the back face. CFD development of three-dimensional viscous computer codes with finite rate chemistry are used to compute the mixing and reaction for these devices. Shown in a video is also an unswept configuration for comparison. The computations performed by Moon at LeRC illustrate the extent of the reacting zone for the two configurations. It should be noted that the CFD developed for the combustor does not truly represent the turbulence-chemistry interaction. The chemistry is modeled using a number of chemical steps (12) and a number of species (9). Mean values of temperature and pressure are used to determine the reaction rates. Current research is devoted toward formulating a probability density function model for the chemical reactions. Such a scheme would rely on local instantaneous values of temperature and pressure for the chemical reaction calculations. Again, we shall hear in this session, some further discussion on the mixing and combustion issue.

IV. EXHAUSTING THE AIR

The nozzle, like the inlet, blends into the aerodynamic lines of the vehicle. Here, the underside of the aft portion of the airplane forms a one-sided nozzle surface. On the opposite wall, a short cowl allows the flow to form a free shear layer with the external flow field. Hence, the nozzle dynamics and the shear layer physics and chemistry are radically different than those within our experience. Vehicle speed affects the effective back pressure on the nozzle and causes it to be over- or under-expanded. Shock-shear layer structure is dramatically affected, and can vary from shock impingement on the vehicle to no effect. The composition of the species entering the nozzle and the exit conditions influence the completeness of chemical reaction in the plume. A typical computation by Lai at LeRC using a Reynolds averaged, three-dimensional Navier-Stokes codes is shown in the video. This computation relies on a Baldwin-Lomax turbulence model. One can see the development of sidewall shear layers at the nozzle exit as well as the corresponding features on the cowl surface and shear layer. The nature of the exhaust plume is highly affected by three dimensionality. Only limited data exist for flow field comparison at the present time. There is no doubt, however, that significant closure issues remain to be addressed.

V. CONCLUDING REMARKS

On an overall basis, one can observe that a significant amount of progress has been made on the application of CFD for high-speed airbreathing propulsion systems. Excellent qualitative agreement is the usual picture, with significant discrepancies only in those near wall regions dominated by strong viscous flows. Nonequilibrium air effects and finite rate chemistry are extensively modeled and computed. However, proper turbulence chemistry interaction requires a significant amount of attention. Improvements in turbulence and transition models are also critically needed. I look forward to hearing the presentations in this session on these important issues; I hope you share my enthusiasm.

NUMERICAL SIMULATION OF FLOW THROUGH OPPOSITE AND SIMILAR SWEEP SCRAMJET INLETS

D. J. Singh
Analytical Services & Materials, Inc.
107 Research Drive,
Hampton, VA 23666 U.S.A.

and

Ajay Kumar
NASA Langley Research Center
M.S. 156
Hampton, VA 23665 U.S.A.

Abstract

A comparative numerical study of performance parameters of a similar and an opposite sweep sidewall compression inlet is made. A three-dimensional Navier-Stokes code is used to calculate the flow through these inlets. Results of these calculations are used to compare the two designs for their performance and flow quality. Effects of boundary-layer ingestion on the performance and overall flow features are also investigated.

Introduction

For over two decades, NASA Langley Research Center has been conducting research in developing a viable air-breathing propulsion system for hypersonic flight application. In this flight regime, a supersonic combustion ramjet (scramjet) engine becomes attractive. The inlet of the engine module compresses the flow with the swept, wedge-shaped sidewalls. The sweep of these sidewalls, in combination with the aft placement of the cowl on the underside of the engine, allows for efficient spillage and for good inlet starting characteristics over a range of operating Mach numbers with fixed geometry [1]—[2]. In order to systematically investigate the effects of sweep on the performance of a scramjet inlet, a numerical study is conducted on two equivalent scramjet inlets. These inlets have been designed in such a way that both have the same wetted area [3] but in one inlet design, all of the compression surfaces are swept backward (Fig. 1-a); whereas, in the other design, alternate surfaces are swept backward and forward (Fig. 1-b). A three-dimensional Navier-Stokes code, SCRAMIN [2], is used to analyze the inlet configurations. The code solves the Reynolds-averaged Navier-Stokes equations in conservation form using the MacCormack method.

Results and Discussion

The numerical simulation of the flow through the two inlet configurations described earlier is made for the following freestream conditions

$$M_\infty = 4.5, \; T_\infty = 200°K, \; p_\infty = 3376.86 \; N/M^2$$

The calculations were made for uniform flow entering the inlet configurations and with a 10% (of inlet height) and a 20% thick entering boundary layer to determine the effect of boundary-layer ingestion on the performance of the similar and opposite sweep inlet. Due to space limitation, only representative results are shown here; the detailed results are available in Ref. [3].

Figure 2 shows the pressure plots in three longitudinal planes for the two configurations with uniform flow entering the inlet. It shows the shock and expansion waves and their interactions. This figure shows an advantage of the opposite sweep over the similar sweep. The sidewall shocks in the opposite sweep inlet do not intersect with the sidewall boundary layer in a given swept, constant-area cross-section. Therefore, any blockage created by the shock-induced separation of the boundary layer is not as damaging to the inlet performance as in the case of similar sweep inlet where the shock/boundary layer interaction and associated separation takes place in a swept, constant-area cross-section.

Figure 3 shows plots of Mach number, stagnation pressure, and static pressure in a cross plane near the throat for the two configurations with 20% thick boundary layer entering the inlets. It shows approximately half of the cross-section is now filled with the viscous, nonuniform flow because the 20% thick entering boundary layer near the top wall is being squeezed into the throat region which is four times smaller in width than the entering cross-section.

Using the flowfield results, calculations were also made for inlet performance quantities such as the average throat Mach number, total pressure recovery, axial thrust, and mass capture. The calculations showed that there was little difference in the average throat Mach number and total pressure recovery of the two inlets. However, the mass capture significantly increased for the opposite sweep inlet. Thus, the detailed flowfield results suggest that the overall impact of the opposite sweep is quite favorable on the inlet performance.

References

1. Trexler, Carl A, "Inlet Starting Predictions for Sidewall Compression Scramjet Inlets," AIAA Paper No. 88-3257, July 1988.
2. Kumar, Ajay, "Numerical Simulation of Scramjet Inlet Flow Fields," NASA TP-2517, May 1986.
3. Kumar, Ajay, Singh, D. J., and Trexler, C. A., "A Numerical Study of the Effects of Reverse Sweep on a Scramjet Inlet," AIAA Paper No. 90-2218, July 1990.

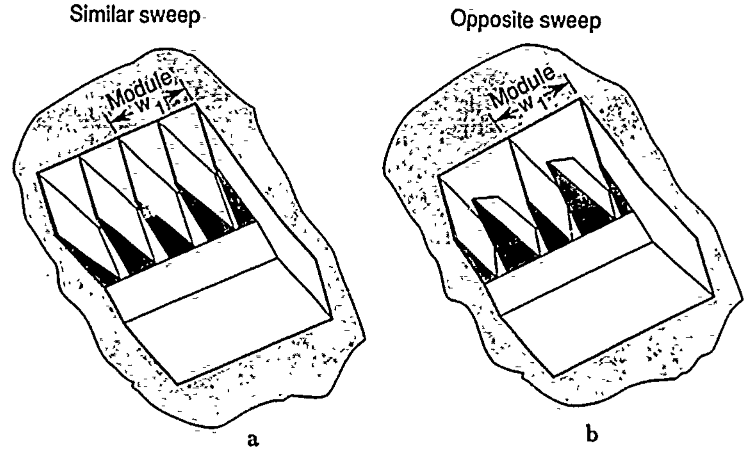


Figure 1: Generic inlet configuration for the study of sweep effects.

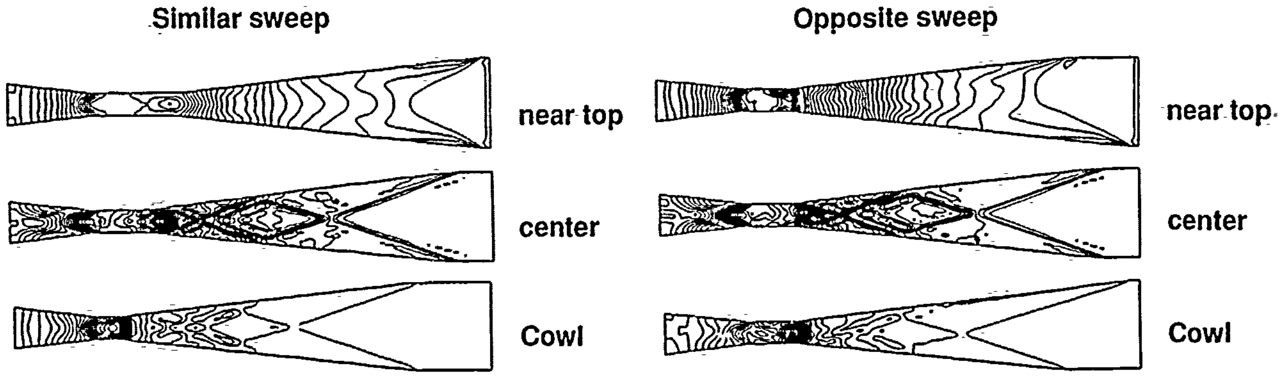


Figure 2: Static pressure contours at three height locations with uniform entering flow.

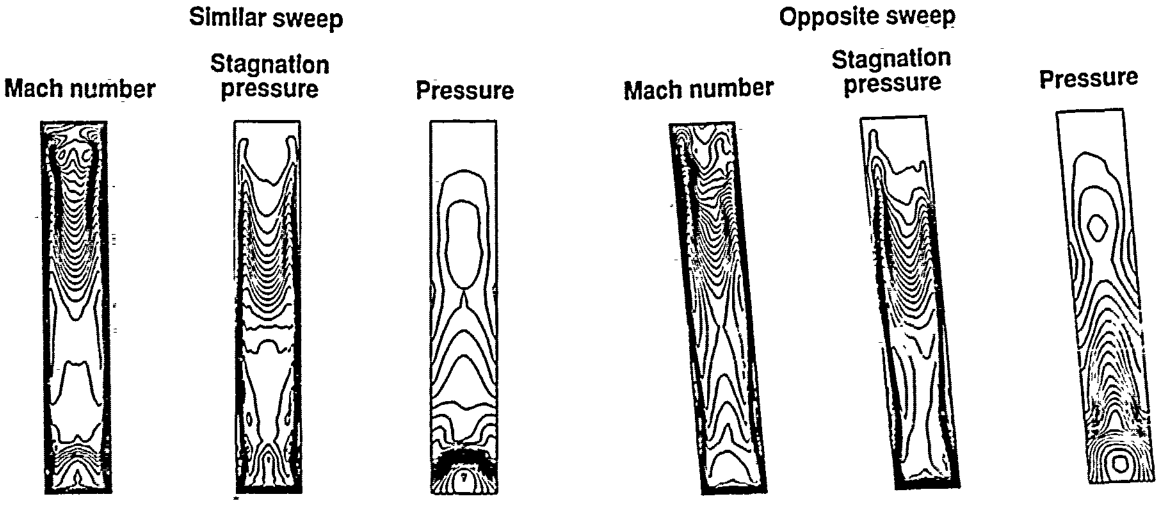


Figure 3. Plots of inlet performance parameters near throat with 20% thick entering boundary layer.

NUMERICAL STUDY ON SUPERSONIC CHEMICALLY REACTING FLOWS

SATORU OGAWA, YASUHIRO WADA AND TOMIKO ISHIGURO
Computational Sciences Division, National Aerospace Laboratory, Chofu, Tokyo, Japan

Abstract This paper describes the numerical techniques to solve the supersonic chemically reacting flows. The higer-order upwind scheme based on a generalized approximate Roe's Riemann solver is used, and a fully implicit time integration method is used to accelerate convergence rates. As the numerical examples, the hypersonic flow around space vehicle, and the supersonic combustion in SCRAM jet engine are presented.

I. INTRODUCTION

The basic research and development is driven forward in the NAL for the space plane which are capable to go into the space and return with ease. The most important subjects in this development would be, the precise evaluation of aerodynamic and aerothermodynamic characteristics in hypersonic region where the real gas effect is dominant, and the development of the supersonic combustion RAM (SCRAM) jet engine. No ground-based experimental facilities can fully duplicate the conditions that these vehicles will encounter in the upper atmosphere, hence the numerical simulation is expected to be one of the most promising method in the study of hypersonic chemically reacting flows. In this study, chemical nonequilibrium flows are solved by the use of higher-order upwind scheme. This scheme[1] is based on a generalized Roe's approximated Riemann solver, and arbitrary nonequilibrium effects are treated in a unified formulation. A fully implicit time integration method is used to accelerate convergence. As numerical examples, chemically reacting hypersonic flows around the Space Shuttle, and the supersonic combustion in the SCRAM jet engine are solved.

II. BASIC EQUATIONS

The basic equations of compressive chemically reacting flow are written in the weak conservation form:

$\partial q_i / \partial t + \partial F^k{}_i(q) / \partial x^k = s_i$,

where q_i is a conservative quantities per unit mass, and s_i is its corresponding source term.

$q = [\rho, \rho u, \rho v, \rho w, E, \rho Y_1, \rho Y_2, \cdots, \rho Y_n]^T$,

where E is the total enrgy, $E = e + 1/2 \rho V^2$, and the internal energy e is given by $e = \rho \sum Y_i [\int Cp_i dT + \Delta HF_i] - P$. Y_α, ($\alpha = 1, \cdots n$), are the mass fraction of α-speicies, and the Arrhenius type chemical reaction models are used for the source terms of speices conservation equations.

III. NUMERICAL SCHEME

In recent years, upstream difference schemes have yielded a great success in flow computation. Most of these schemes make use of the exact or approximated solution of the Riemann problem as a building block. Among them, Roe's approximated Riemann solver is one of the most promising method, and in this paper, a generalized Roe's approximate Riemann solver for nonequilibrium flows is used. Chakravarthy Osher postprocessing TVD scheme is used to make the higher-order scheme. It is more efficient than MUSCL scheme for a nonequilibrium flow problem, because the latter method needs Newton iterations at each cell interface to calculate the value of temperature from conservative variables in the case of chemically reacting flows.

Generally nonequilibrium flow is very stiff problem, so that it is desirable to treat every term implicitly. In this paper each convective block operator is diagonalized so that the block matrix operation is reduced to the scalar one. Further, in order to enhance robustness, near by shock waves only chemical source terms are implicitly treated and the rest explicitly. This patched method[2] is easily constructed by replacing tri-diagonal scalr operators by a unit matrix.

IV. EXAMPLES OF NUMERICAL COMPUTATIONS

A. Hypersonic flow of real gas

Since the space plane flies more than ten times as fast as the sonic speed, the extremely strong shock wave appears ahead. The strong compression behind the strong shock wave causes the high temperature more than ten thousand degree near the plane. Thus the nitrogen and oxigen in the atmosphere dissociate, consequently the usual assumption of perfect gas no longer holds good and it becomes necessary to include the effect for the real gas. In this example the elemental reactions for 7 components of N_2, O_2, N, O, NO, NO^+, e are considered as the dissociation in order to include this real gas effect. In the case of the space plane, when the shock wave strikes the wing, the plane suffers the very severe aerodynamic heating. Therefore the prediction for the situation of shock wave is very important and the correct evaluation for the real gas effect is necessary. In Fig.1 are shown the numerical solution for the hypersonic flow of Mach number 15.7 around the Space Shuttle with the real gas effect included. The distribution for the mole fraction of the atomic oxigen produced by the dissociation is displayed in the figure. The atomic oxigen near the nose of the plane is transported with fluid to gather to the center parts on both the upper and lower surfaces separately. Further on the upper surface it is transproted with the separated fluid to spread over the plane again.

B. Chemically reacting flow in a combustor

As the computation of chemically reacting flow has been possible with the progress of computers, the computational condition of this example is almost adapted to the actual experimental one for the SCRAM jet engine. The foundation of physical phenomenon in the SCRAM engine is that the hydrogen blows up into the high temperature gas and burns. For the combustion it is necessary to introduce the reacting model. In the Westbrook reacing model 9 chemical species of N_2, H_2, O_2, OH, H_2O, H, O, H_2O_2 and HO_2 are considered and 17 elementary reaction steps are contained. The chemically reacting flows in the combustor where the hydrogen blows up have been numerically solved using the TVD scheme for the governing equation system above stated. An example is shown in Fig.2 , where iso-Mach contours are shown. Thus by the CFD it is possible to see the flow details such as the Mach disk formed by the blow, which is difficult to measure by experiments. Since the problem of turbulence in reacting flows remains almost unsolved, however, the reacting ratio does not show a good agreement between the numerical solution and the experiments. The reason is that the reacting ratio greatly depends on the mixing of hydrogen and oxigen, which the turbulent diffusion governs. The problem of turbulence in reacting flows would be a important researching theme in the future.

V. CONCLUDING REMARKS

By the development of CFD till now, it has been possible to obtain the numerical solutions which hold good to some extent for a large variety of problems. The problems for not only chemical reaction but also radiation and electro-magnetic fluid dynamics can be numerically solved without difficulty if costing much time. It may fairly be said that the final problem still remained is the turbulence that is a remarkable characteristics of the non-linear fluid motions.

References

[1] Wada,Y, et al., AIAA Paper 88-3596Cp, 1988.
[2] Wada,Y, et al., AIAA Paper 89-0202, 1989.

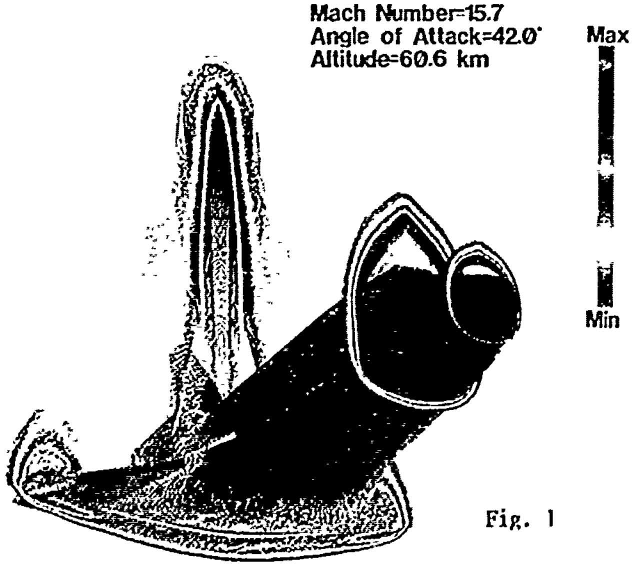


Fig. 1

MOLE FRACTION OF OXYGEN ATOM

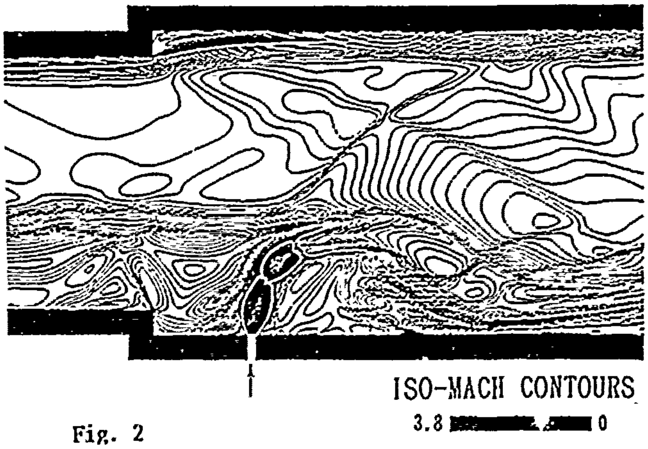


Fig. 2

RAPID PHASE CHANGES IN ISO-OCTANE IN THE GENERAL VICINITY OF THE CRITICAL POINT

S. C. Gulen, H. J. Cho, A. Hirsa, P. A. Thompson, M. Moran

Department of Mechanical & Aerospace Engineering
Rensselaer Polytechnic Institute
Troy, NY, U. S. A.

Abstract

Stationary states at high temperatures and pressures in Iso-Octane (2-2-4 Trimethylpentane) were achieved behind a compression shock wave reflected from the shock-tube end wall. The end states cover a broad region including vapor, liquid, and mixture phases. Measurements of shock velocity, pressure, and temperature were performed together with extensive photographic observation of the final states through a sapphire window mounted at the end of the shock-tube observation chamber. In general, phase changes resulting from the shock compression of a retrograde substance are predicted reasonably well with the equilibrium Rankine-Hugoniot model. Photographic observations of the end states show a rich variety of two-phase vortex rings, depending on initial conditions and shock strength.

Introduction

Adiabatic, pressure driven, finite amplitude waves resulting in phase changes - from vapor to liquid or liquid to vapor - have been observed in retrograde fluids. Unlike thermally driven phase transitions in regular fluids such as water, these waves result from a jump in pressure. Recent work on the subject includes complete and partial liquefaction shocks [1], rarefaction shock from a critical state [2], shock splitting [3], and mixture evaporation rarefaction shock [4]. In this work rapid phase change phenomena in the general vicinity of the critical point are investigated. Thermodynamic, vapor-liquid critical point as described by modern power laws [5] is a singular point of infinite isothermal compressibility and constant volume specific heat (hence, in principle, zero soundspeed.) A recent study of nonequilibrium, near-critical states in shock tube experiments revealed a minimum in the soundspeed (about *10 m/s*) and disappearance of two distinct phases at a pressure ca. *25%* above the critical value [6].

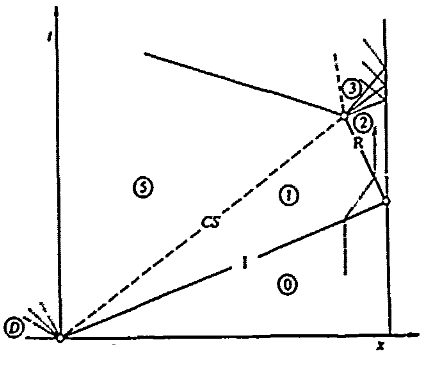


Figure 1. *x-t* diagram of the shock-tube flow. The closed end of the test section is at right. Arabic numerals designate test fluid states. I=incident shock, R=reflected shock, CS=Contact surface, D=Driver gas initial state. Not to scale.

Theory & Results

The *x-t* diagram of the reflected shock system is shown in Figure 1. Figure 2 shows a detailed reflected shock adiabat on a *P-v* surface. The adiabat crosses the phase boundary but the process is so fast that the condensation is delayed until the limit of supersaturation, Wilson line, is reached whereupon a spontaneous collapse of the metastable state occurs. If the shock strength is such that the end state lies below point 2 (triple point) two distinct discontinuities are observed: a forerunner, "dry" shock and a condensation discontinuity [3]. In our experiments the shock Mach numbers are high enough that the end states lie above the triple point and a single liquefaction shock front exists.

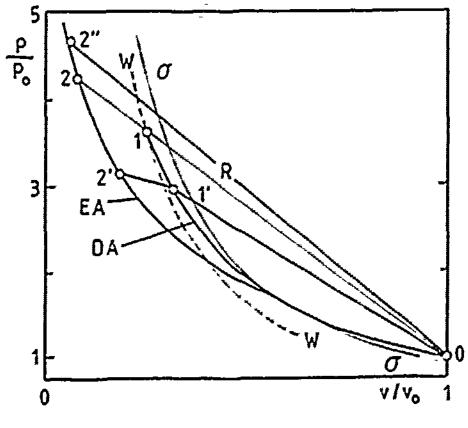


Figure 2. Pressure-volume diagram of the reflected shock. EA=equilibrium adiabat (liquefaction shock); DA=dry adiabat (non-equilibrium, metastable, supersaturated vapor); R=Rayleigh line; σ=saturated vapor boundary; W=Wilson line (line of critical supersaturation); 2=triple point (see [3] for details.)

In the calculations Rankine-Hugoniot equation is used together with a virial-type, corresponding states equation of state which is modified to represent the near-critical region more accurately. Figure 3 shows experimental data for a reflected shock system which passes through the theoretical critical point. The agreement with the calculations is quite good. Nonetheless, there is a systematic deviation in temperatures measured by very thin (*$5x10^{-4}$ in* diameter), fast response thermocouples. They are lower than calculated values by ca. *5* to *10°C* depending on the shock strength.

In addition to the measurement of temperature and pressure, photographic observations are made for different incident shock Mach numbers. Four different initial conditions are chosen with resulting reflected shock adiabats shown in Figure 4. Particular emphasis was given to the end states where the shock adiabat crosses saturated vapor and liquid boundaries. Designation of these regions are as shown in Figure 4 and a description of the nature of the associated phase change phenomena is given in Table 1. Except from the region on the saturated vapor boundary far from the critical point (Figure 5 a), two-phase vortex rings are consistently observed both on saturated vapor and liquid boundaries. The vortex rings are thought to be a direct result of steep pressure and density gradients across the shock wave acting upon the nucleation clusters. In regions *SL1* or *SL2* (Figure 5 b&c, respectively), a group of large, turbulent vortex rings amongst thousands of tiny, small-scale, newly born vortex rings have been observed. Vortex rings which are formed become turbulent within a few tens of microseconds and turbulent vortex rings grow linearly with time (Figure 6.) Vortical structures near the critical point (Figure 5 d&e) are even more turbulent and diffuse. It is important to note that the critical point we are referring to is the one calculated by the equation of state. The proximity of the observed states to the *actual* , non-equilibrium critical point is open to discussion.

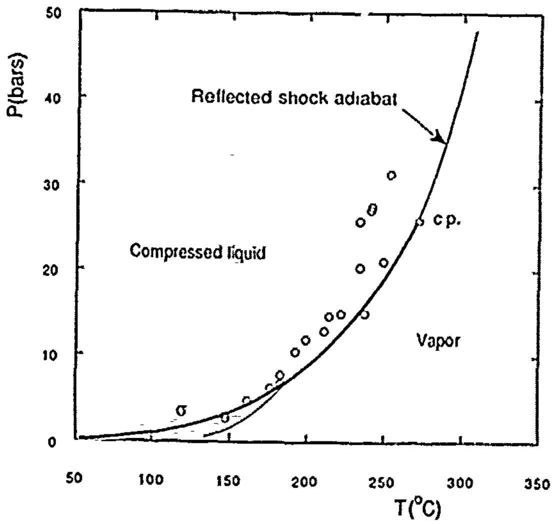


Figure 3. Pressure and temperature behind reflected shock. $T_0=130$ °C, $P_0=0.445$ *bars*. σ = vapor pressure line, *c.p.* = critical point.

Unique and fascinating structures are observed in region *SV3* (Figure 5 f.) These are circular, two-phase structures with apparent *rays* radially emanating from the center. As the end states approach the critical point, i.e. regions *SV4* and *SV5* (Figure 5 g&h, respectively), these ray structures still can be seen.

The structure of the liquefaction shock front can be divided into two parts. A frozen discontinuity dominated by the viscosity and thermal conductivity, typically a few mean free paths wide, followed by a considerably longer relaxation zone where nucleation and droplet growth take place. The compression rate across the frozen part of the shock is so high that the nucleation process lags. There are mainly three relaxation processes, inertial re'axation (there is a velocity slip between droplets and suddenly decelerated vapor), droplet temperature relaxation, and vapor thermal relaxation. To give a numerical example, an incident shock ($M_0=2.46$) produces a two-phase mixture (quality $x=0.33$) after reflection from the end wall. The width of the frozen discontinuity is computed to be approximately $0.05\ \mu m$ (ca. 5 mean free paths) with a resulting supersaturation of $S=2.3$.

Shock Adiabat	States behind the reflected shock	Description of phase change
A	Liquid Saturation Boundary : SL1 Vapor Saturation Boundary : SV1	Well defind two - phase vortex rings in liquid or dense gas Condensation in a normal fashion in vapor
B	Liquid Saturation Boundary : SL2 Vapor Saturation Boundary : SV2	Well defind two - phase vortex rings in liquid or dense gas Condensation in a normal fashion in vapor
C	Near Critical Region : NC Vapor Saturation Boundary : SV3	Feathery, more diffuse and turbulent two - phase vortex rings Interesting ray structure in condensation in vapor
D	Vapor Saturation Boundary : SV5 Vapor Saturation Boundary : SV4	Ray structure, two - phase vortex rings in vapor Ray structure, two - phase vortex rings in vapor

Table 1. Summary of phase changes on the vapor and liquid saturation boundaries.

Approximate calculations showed droplet temperature relaxation (ca. $4x10^{-15}$ *sec.*) to be much faster than inertial relaxation which is itself approximately one order of magnitude faster than vapor thermal relaxation ($7x10^{-10}$ and $1x10^{-9}$ *sec.*, respectively.) Numerical integration of the equations of motion coupled with a suitable nucleation and droplet growth model will yield more accurate description of the liquefaction shock structure.

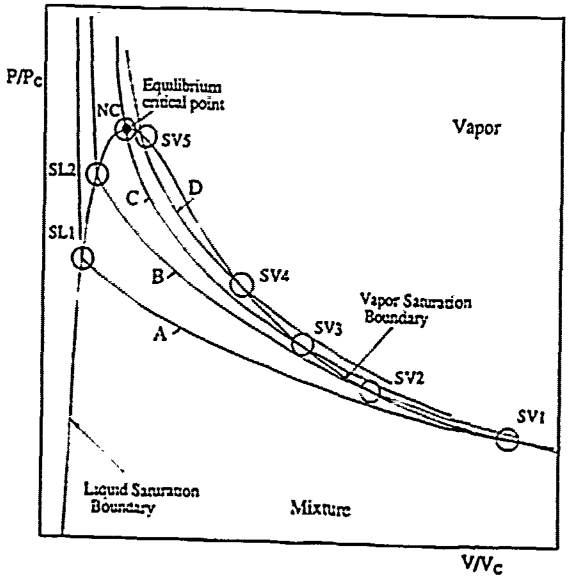

P/Pc
Equilibrium critical point
NC
SV5
Vapor
SL2
C
D
SL1
B
SV4
Vapor Saturation Boundary
SV3
A
SV2
SV1
Liquid Saturation Boundary
Mixture
V/Vc

Figure 4. Four reflected shock adiabats *A*, *B*, *C*, and *D* along which experiments are done with increasing shock Mach numbers. *SL1*, *SL2*, *SV1*, *SV2*, *SV3*, *SV4*, *SV5*, and *NC* designate regions where extensive photographic observations are made (see also Table 1.) Drawing is not in scale.

Conclusion

Photographic observations show a variety of new and rich phenomena associated with rapid phase changes across the liquefaction shock front. Two-phase vortex rings are formed by the pressure gradient across the shock front acting upon the nucleation sites which are denser than the surrounding gas. Vortex rings quickly become turbulent upon formation and grow linearly thereafter. Measured downstream pressures agree well with equilibrium calculations, whereas temperature measurements are lower than computed values. The results are mainly qualitative and current efforts focus on quantitative description of the vortex formation and liquefaction shock structure.

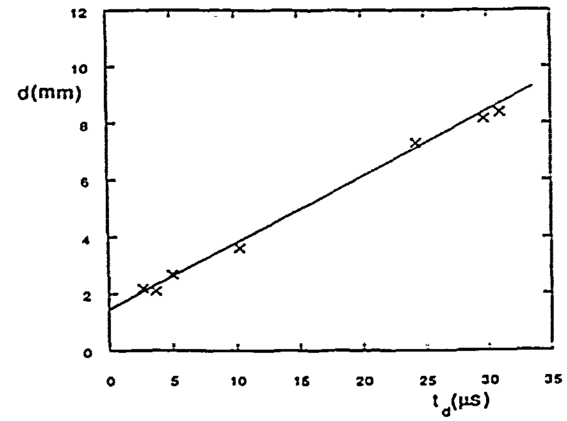


Figure 6. Average vortex ring diameter versus time. Vortex rings are produced in Region *NC*. *T_0=130°C, P_0=0.445 bars.*

References

1. Dettleff, G., Thompson, P. A., Meier, G. E. A., Speckmann, H.-D., *J. Fluid Mech.*, 95, 279-304, 1979.

2. Borisov, A. A., Borisov, Al. A., Kutateladze, S. S., Nakoryakov, V. E., *J. Fluid Mech.*, 126, 59-73, 1983.

3. Thompson, P. A., Chaves, H., Meier, G. E. A., Kim, Y.-G., Speckmann, H.-D., *J. Fluid Mech.*, 185, 385-414, 1987.

4. Thompson, P. A., Carofano, G. C., Kim, Y.-G., *J. Fluid Mech.*, 166, 57-92, 1986.

5. Bejan, A., *Advanced Engineering Thermodynamics*, Wiley Interscience, New York, pp. 299-338, 1988.

6. Thompson, P. A., Kim, Y.-G., Yoon, C. J., Chan, Y., *Proceedings of the 16th Int. Symp. on Shock Tubes & Waves*, Aachen, Germany, July 26-31, 1987.

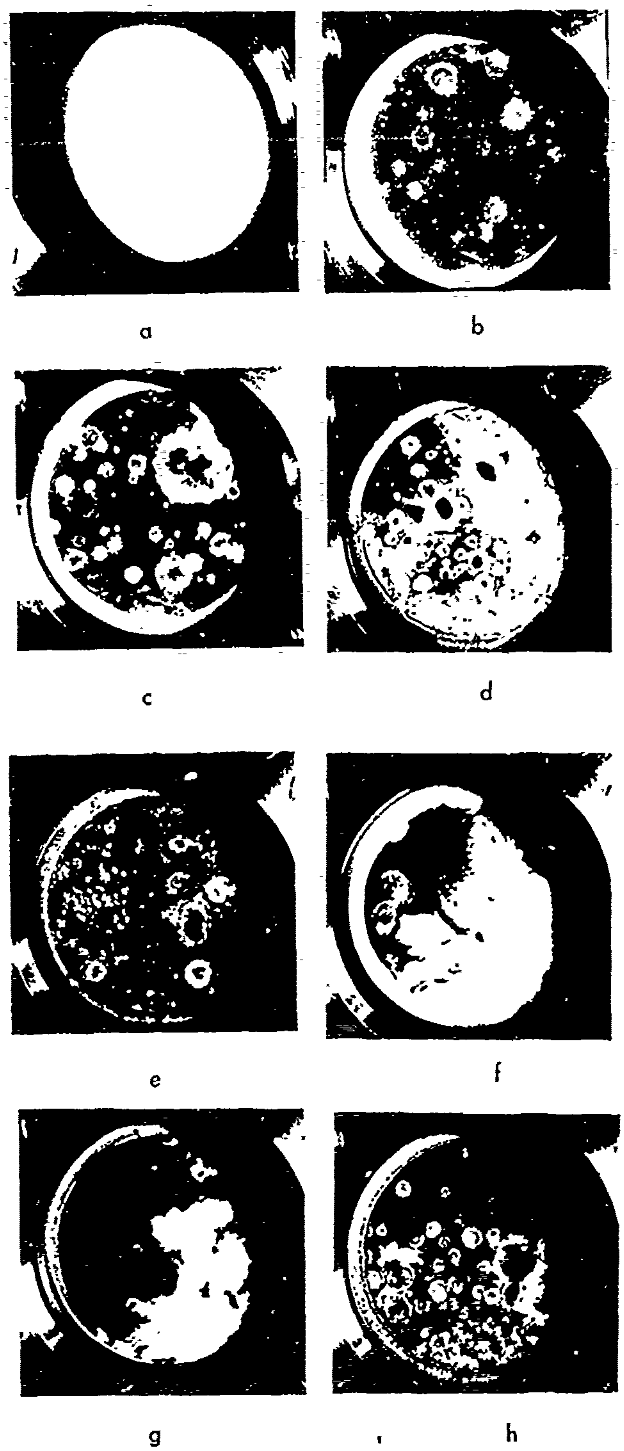


Figure 5. Photographs taken during the rapid phase change experiments through a sapphire window mounted at the end wall of the shock-tube. See Figure 5 and Table 1 for region designations. a) Region *SV1*, *M_0=2.46*; b) Region *SL1*, *M_0=2.94*; c) Region *SL2*, *M_0=2.85*; d) Region *NC*, *M_0=2.82*; e) Region *NC*, *M_0=2.87*; f) Region *SV3*, *M_0=2.30*; g) Region *SV4*, *M_0=2.33*; h) Region *SV5*, *M_0=2.78*.

TRANSONIC FLOWS OF BZT FLUIDS

M. S. CRAMER
Department of Engineering Science and Mechanics
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061–0219 U.S.A.

Abstract — We examine steady transonic flows of Bethe–Zel'dovich–Thompson (BZT) fluids. An extension of the transonic small disturbance equation,valid in the neighborhood of one of the zeros of the fundamental derivative, is presented. Numerical solutions reveal that the natural dynamics of these fluids may result in a significant increase in the critical Mach number. We also report transonic flows involving both expansion and compression shocks in the same flowfield.

I. INTRODUCTION

Transonic flows are inherently nonlinear and are typically accompanied by shock formation in the hyperbolic portions of the flow. Because of the strong adverse pressure gradients associated with compression shocks, shock–induced separation is a major concern in the design of transonic turbomachinery. Such separation is a major loss and vibration mechanism. The process of shock formation is due to the intrinsic nonlinearity of the fluid. As a result, it has been assumed that all flows behave generally the same as predicted by the perfect gas theory and most work has addressed blade design to minimize these effects.

The appropriate measure of the intrinsic nonlinearity of the fluid is the thermodynamic parameter

$$\Gamma \equiv \frac{V^3}{2a^2}\frac{\partial^2 p}{\partial V^2}\bigg|_s, \qquad (1)$$

where V, p and s are the fluid specific volume, pressure and entropy. The quantity

$$a \equiv V\left[-\frac{\partial p}{\partial V}\bigg|_s\right]^{1/2}, \qquad (2)$$

is the thermodynamic sound speed. Here we follow Thompson [1] in referring to (1) as the fundamental derivative of gasdynamics. Inspection of (1) indicates that Γ is a measure of the curvature of the isentropes.

Recent studies have shown that the gasdynamics of fluids having relatively large specific heats may be qualitatively different than that of lighter substances such as air and water. The main objective of the present study is to examine the behavior of these fluids in the transonic regime. In particular, it will be shown that use of these fluids results in significant increases in the critical Mach number, thereby decreasing the range flow speeds at which the undesirable transonic flow effects are observed.

The fluids of interest here are those for which $\Gamma < 0$ for a finite range of pressures and temperatures. The conditions under which $\Gamma < 0$ where first given by H. A. Bethe [2] and Ya. B. Zel'dovich [3], who demonstrated that fluids having relatively large specific heats will have a region of negative Γ in the fluid's dense gas regime. The general region where $\Gamma < 0$ is depicted in Figure 1 where the isentropes of normal decane have been computed and plotted on a p–V diagram. The regions of downward curvature correspond to $\Gamma < 0$. Further examples of commonly encountered fluids having $\Gamma < 0$ have been provided by Thompson and co–workers [5], [6], and Cramer [7]. Because of the contribution of the early workers in this area, we refer to fluids possessing a region of $\Gamma < 0$ in the single–phase regime as Bethe–Zel'dovich–Thompson (BZT) fluids.

The significance of BZT fluids is that the well–known compression shocks of the perfect gas theory violate the entropy inequality and disintegrate into centered fans if $\Gamma < 0$. Expansion shocks, normally forbidden in the perfect gas theory, not only form from smooth waves but are seen to satisfy the entropy inequality in flows having $\Gamma < 0$ everywhere. It therefore appears that the natural dynamics of BZT fluids may lead to a reduction or even elimination of shock–induced separation.

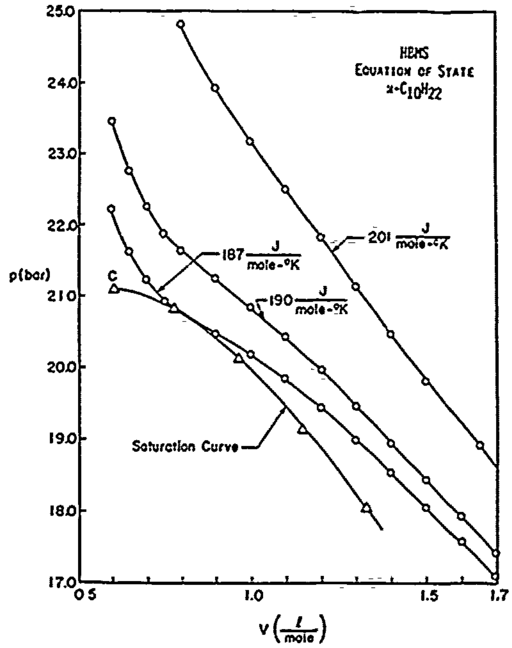


Figure 1. Computed isentropes for n–decane. Gas model is the HBMS [4] equation of state.

Thus, there is likely to be advantages in the use of BZT fluids even in supercritical flows. For further details of the remarkable dynamics of BZT fluids, we refer the reader to the surveys found in References [1], [8]–[9].

II. THE TRANSONIC SMALL DISTURBANCE EQUATION

We consider small two–dimensional disturbances to a near–sonic flow of a BZT. The usual assumptions of the transonic small disturbance theory will be supplemented by the condition that the thermodynamic state of the freestream is in the vicinity of one of the zeros of the fundamental derivative. Such zeros form the boundary between the positive and negative Γ regions and are recognized as the inflection points of the isentropes in Figure 1. With this freestream state the small perturbations caused the thin blade can take the flow from regions of positive to negative Γ and vice versa. In this sense, the flow will be qualitatively similar to those involving larger amplitudes. Furthermore, the most complex and interesting features occur when the nonlinearity is mixed. When these assumptions are incorporated, Cramer [8] has shown that the extension of the classical transonic small disturbance equation is

$$\left[M_\infty^2 - 1 + 2\Gamma_\infty\phi_x - \Lambda\phi_x^2\right]\phi_{xx} = \phi_{yy}, \qquad (3)$$

with the usual boundary conditions at the blade and at infinity. Here ϕ is a nondimensional velocity potential, x is the scaled distance in the freestream direction and y is the scaled distance normal to the freestream.

The Mach number and fundamental derivative in the freestream are denoted by M_∞ and Γ_∞, respectively. The parameter

$$\Lambda \equiv \rho \frac{\partial \Gamma}{\partial \rho}\Big|_s , \tag{4}$$

where $\rho \equiv V^{-1}$ is the fluid density, is a second nonlinearity parameter. This is usually evaluated at the state corresponding to the zero in the local value of Γ, without loss of accuracy to the overall approximation. Near the large density zero of Γ, typical values of Λ are estimated to be 1.5–2.5. This second nonlinearity parameter is negative near the low density zero of Γ.

III. RESULTS

Numerical solutions to (3) have been generated through use of an extension of the Murman–Cole scheme. The boundary conditions were those of a circular airfoil. The results of these calculations are depicted in Figures 2–4. In order to check the scheme, calculations for a classical (Λ=0) case were carried out. Here $\Gamma_\infty = 0.4$, $M_\infty = 0.9$ and the half–thickness of the circular arc airfoil was taken to be 0.06 of the chord The flow is from left to right. Scaled values of the pressure coefficient are plotted in Figure 2. Here it is seen that the given Mach number is considerably above the critical value with the shock well back on the wing.

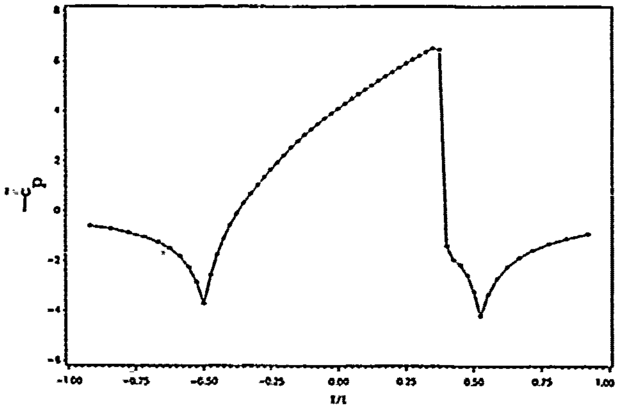


Figure 2. Scaled pressure coefficient $\bar{c}_p$ over a circular arc airfoil. $\Gamma_\infty = 0.4$, $\Lambda = 0$, $M_\infty = 0.9$ and the thickness is 0.06 of the chord.

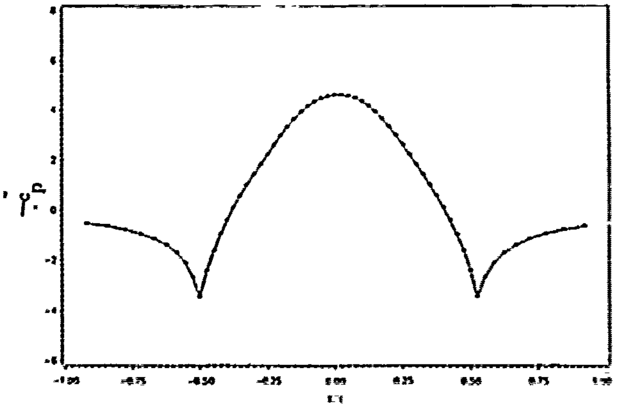


Figure 3. Scaled pressure coefficient $\bar{c}_p$ over a circular arc airfoil. All conditions are identical to those of Figure 2 except that $\Lambda = 1.0$.

To determine the influence of the new term in (3) we then examined the same case with $\Lambda = 1$. Because $\Lambda > 0$ the undisturbed state is near one of the high pressure zeros of Γ, this point corresponds to the high pressure inflection points on the isentropes of Figure 1. The results of these calculations are found in Figure 3. Here it is seen that the large compression shock has entirely vanished.. A detailed examination of the Mach numbers indicates that the flow remains entirely subsonic That is, the use of a BZT fluid in the neighborhood of one of its zeros have driven the critical Mach number to values above 0.9. Physically, the increase is due to the fact that the expansions over the top of the blade has shifted the flow into the $\Gamma < 0$ regime before the sonic state is reached. As pointed out in References [1], [8], and [10] acceleration through the sonic condition is then difficult, if not impossible. In order to save space, we refer the reader to these previous studies for a detailed account.

The increase in the critical Mach number is by no means an isolated case. Further detailed studies suggest that blade configurations giving rise to a critical Mach number of 0.69 in air may correspond to a critical Mach number over 0.98 in many of the BZT fluids described in Reference [6] or [7].

As a final example, we have retained the same upstream thermodynamic state as used in Figures 2–3, but have raised the Mach number to 0.93, approximately. In this case, the flow is able to achieve sonic conditions before the $\Gamma = 0$ point is reached. The result is plotted in Figure 4. In this supercritical flow, two shocks appear. The first (leftmost) is an expansion shock which is followed by a compression shock. This is in marked contrast to classical theory where no more than one shock can occur.

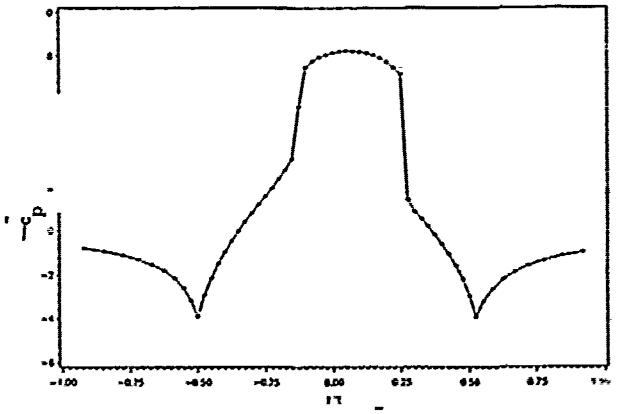


Figure 4. Scaled pressure coefficient $\bar{c}_p$ over the circular ar airfoil of Figures 2 and 3. $\Gamma_\infty = 0.4$, $\Lambda = 1.0$ and $M_\infty = 0.93$.

IV. SUMMARY

The preceding has introduced a modified form of the transonic small disturbance equation valid when the freestream is in the vicinity on one of the zeros of the fundamental derivative of a BZT fluid. The numerical results demonstrate that the natural dynamics of BZT fluids can give rise to significant increases in the critical Mach number as well as qualitative differences in the details of supercritical flows.

The author would like to thank G. M. Tarkenton for providing the plots of Section 3, and Dr. H. A. McGee and students for providing the code for the generation of Figure 1. This work was supported by the National Science Foundation under grant #CTS–8913198.

REFERENCES

1. Thompson, P. A., Phys. Fluids, 14, 1971, pg. 1843.
2. Bethe, H. A., Off. of Sci. Res. & Dev. Rep. 545, Wash., D.C., 1942.
3. Zel'dovich, Ya. B., Zh. Eksp. Teor. Fiz., 4, 1946, pg. 363.
4. Hirschfelder, J. P., Buehler, R. J., McGee, H. A., and Sutton, J. R., Ind. Engng. Chem, 50, 1958, pp. 386.
5 Lambrakis, K. C. and Thompson, P. A., Physics Fluids, 5, 1972, pg. 933.
6. Thompson, P. A. and Lambrakis, K. C., J. Fluid Mech., 60, 1973, pg. 187.
7. Cramer, M. S., Phys. Fluids A, 1, 1989, pg. 1894.
8. Cramer, M. S., Article in Nonlinear Waves in Real Fluids, ed. A. Kluwick, Springer–Verlag, 1991.
9. Menikoff, R. and Plohr, B., Rev. Mod. Phys., 61, 1989, pg. 75.
10. Cramer, M. S. and Best, L. M., Phys. Fluids A, 3, 1991, pg. 219.

A NEW HIGH-RESOLUTION GLOBAL SPECTRAL MODEL FOR MEDIUM-RANGE NUMERICAL WEATHER PREDICTION

CLIVE TEMPERTON

European Centre for Medium-Range Weather Forecasts
Shinfield Park, Reading, Berkshire RG2 9AX, U.K.

Abstract. A new high-resolution spectral model has been developed for the production of medium range forecasts at ECMWF. Considerable efficiency gains were required to make this model operationally practicable, these were realized principally by the introduction of a reduced computational grid and a semi-Lagrangian integration scheme.

Current plans for the operational ECMWF spectral model (Simmons et al.,1989) include a doubling of the horizontal resolution from T106 (a triangular truncation at wavenumber 106) to T213, with a corresponding increase in the number of vertical levels from 19 to 31.

Two new ingredients of the numerical integration procedure are essential in order to produce timely operational forecasts with this high-resolution model, given the constraints of the present computer system.

The first is the use of a reduced Gaussian grid (Hortal and Simmons,1991) for the computation of nonlinear terms; in this grid, the number of points per latitude row is decreased towards the poles so that the east-west gridlength remains approximately constant. The computation per timestep is thereby reduced by around 25%, with very little impact on the resulting forecast.

The second ingredient is the introduction of a semi-Lagrangian semi-implicit time-integration scheme, which overcomes the stability criterion of the conventional Eulerian treatment of advection (spectral in the horizontal, finite difference in the vertical). The semi-Lagrangian formulation is basically similar to the fully three-dimensional interpolating version of Ritchie (1991), there are however additional complications resulting from the use of a hybrid vertical coordinate rather than the sigma-coordinate of Ritchie's model. On the other hand, the vertical discretization by finite differences and the purely algebraic elimination between variables in the solution of the semi-implicit equations both lead to some simplifications compared with Ritchie's scheme (finite-element vertical discretization and partly analytic elimination). The opportunity was also taken to reduce the number of transforms between spherical harmonic space and the model grid (Temperton,1991). At resolution T213, the Legendre transforms account for only about 10% of the CPU time of the new model.

Tests have shown that the Eulerian version of the T213 31-level model requires a 3-minute timestep to maintain stability, while the semi-Lagrangian version remains stable and accurate with a 20-minute timestep. In addition to the gains already obtained from the use of the reduced grid, the semi-Lagrangian scheme yields a further factor of about 5 in the efficiency of the model. To demonstrate that the semi-Lagrangian version of the T213 31-level model gives essentially the same forecasts as its Eulerian counterpart, Figs. 1 and 2 show the corresponding 3-day forecasts of the 500 hPa height field starting from the operational ECMWF analysis at 12Z on 15th April 1990.

REFERENCES

Hortal, M., and A. J. Simmons, 1991. Use of reduced Gaussian grids in spectral models. Mon. Wea. Rev. 119, in press.

Ritchie, H., 1991: Application of the semi-Lagrangian method to a multilevel spectral primitive equations model. Quart. J. Roy. Meteor. Soc. 117, in press.

Simmons, A.J., D.M. Burridge, M. Jarraud, C. Girard and W. Wergen, 1989. The ECMWF medium-range prediction models - Development of the numerical formulations and the impact of increased resolution. Meteorol. Atmos. Phys. 40, 28-60.

Temperton, C., 1991. On scalar and vector transform methods for global spectral models. Mon. Wea. Rev. 119, in press.

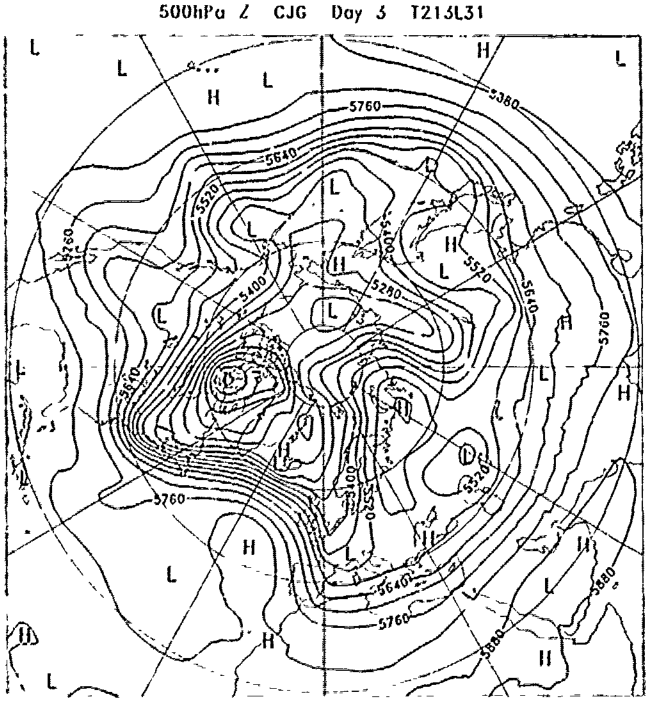


Fig 1. 3-day forecast of 500hPa height field, T213 31-level Eulerian model, 3-minute timestep.

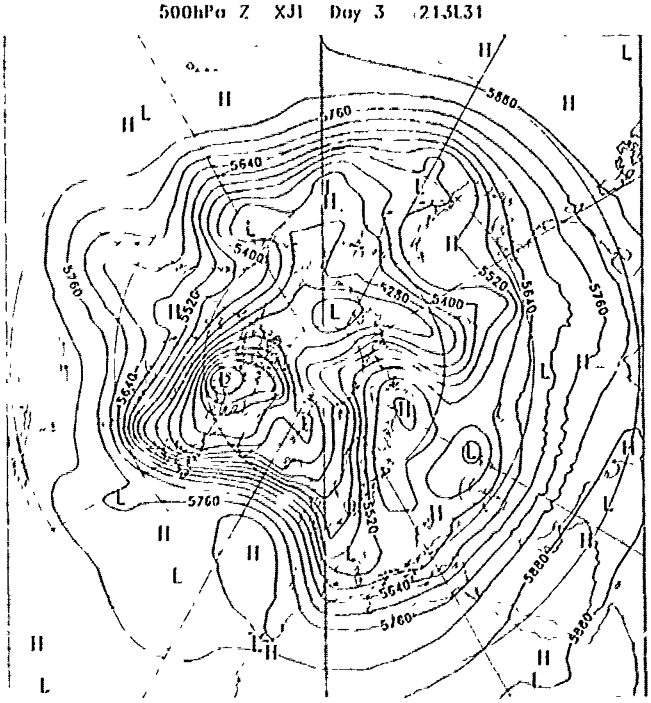


Fig 2 3-day forecast of 500hPa height field, T213 31-level semi-Lagrangian model, 20-minute timestep.

INTEGRATION OF A GLOBAL MULTILEVEL MODEL USING A VECTOR SEMI-LAGRANGIAN SCHEME WITH A MULTIGRID SOLVER

J. R. Bates, S. Moorthi[x] and R. W. Higgins[x]
Global Modeling and Simulation Branch
Code 911
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Abstract

A three-dimensional semi-Lagrangian semi-implicit two-time-level finite-difference integration scheme for the primitive equations of atmospheric motion on the sphere is presented. A trajectory-centered discretization of the governing equations is used, the momentum equation being discretized in vector form before being resolved into components. For the horizontal differencing a C-grid is used, while a Lorenz-grid in σ-coordinates is used in the vertical. The discretized equations, which involve a coupling between all levels, are decoupled by means of a linear transformation, resulting in a set of K (= number of levels) two-dimensional elliptic equations to be solved. With an appropriate formulation of the discretized governing equations, these elliptic equations have a form identical to that encountered in an earlier shallow water model (Bates et al., 1990). The two-dimensional multi-grid solver used in the shallow water case can thus be used to provide an efficient solution for the multilevel case.

A linear stability analysis of the scheme on an f-plane in the absence of a mean flow shows that the scheme is unconditionally stable.

Numerical integrations are performed using an adiabatic version of the model, both with and without orography and divergence damping. The results of these integrations with varying time steps and with both idealized and observed initial conditions will be presented.

Reference

Bates, J. R., F. H. M. Semazzi, R. W. Higgins and S. R. M. Barros, 1990: Integration of the shallow water equations on the sphere using a vector semi-Lagrangian scheme with a multigrid solver. Mon. Wea. Rev., 118, 1615-1627.

[x] Universities Space Research Association

A CLASS OF MONOTONE INTERPOLATION SCHEMES[1]

PIOTR K. SMOLARKIEWICZ and GEORG A. GRELL

National Center for Atmospheric Research[2]
Boulder, Colorado 80307

ABSTRACT

The practice of computational fluid dynamics often requires accurate as well as nonoscillatory interpolation procedures. Such procedures, often referred to as shape-preserving interpolation, may be employed to design monotone advection transport algorithms. This paper poses an inverse problem. A variety of monotone advection schemes with attractive properties has been developed over the last two decades abstracting from any arguments that invoke explicit interpolation procedure. Our goal is to provide a formalism allowing the exploitation of these advection algorithms as shape preserving interpolators. The central theoretical issue concerns a formal equivalence of the advection and interpolation operators on discrete meshes. Through elementary arguments exploiting either the Stokes theorem or the untruncated Taylor formula at 0-th order of expansion, one may show that the solution to an interpolation problem can be expressed as a formal integral of the advection equation. As a consequence, the interpolating operator on a discrete mesh may be represented by an advection scheme, in which the local Courant number vector is replaced by the normalized displacement between a grid point and a point of interest to the interpolation procedure. The accuracy of the resulting interpolation scheme is that of the advection scheme employed.

Among a variety of available advection schemes, the dissipative (forward-in-time) algorithms are the most suitable for practical applications. Since the effective velocity field is constant for every point of interest to the interpolation procedure, these advection schemes retain their formal accuracy of the constant coeffcients limit. Constancy of the effective velocity in an arbitrary-dimensional problem allows for a straightforward alternate-direction implementation of one-dimensional advection schemes without introducing errors characteristic of the time-split advection procedures in variable flows.

Insofar as the linear dissipative advection schemes are concerned, there is no particular gain from such an excercise as the resulting interpolators may be alternatively derived with the help of more traditional arguments invoking either the truncated Taylor formula or Lagrangian polynomial fitting. However, when the preservation of monotonicity and/or sign of the interpolated variable is essential, then the approach adopted becomes useful. For instance, a variety of monotone (and sign-preserving) interpolation schemes of different overall accuracy and complexity levels may be generated using Flux-Corrected-Transport (FCT) versions of high-order-accurate dissipative schemes. The utility of such monotone interpolators is illustrated with examples of applications to selected problems of atmospheric fluid dynamics.

[1] An article of the same title has been submitted to *J. Comp. Phys.*, the preprints are available from the authors upon request.

[2] The National Center for Atmospheric Research is sponsored by the National Science Foundation.

Terrain-following vs. a blocking system for the representation of mountains in atmospheric models

Fedor Mesinger
UCAR Visitor Research Program
National Meteorological Center
Washington, DC 20233, U.S.A.

and

Thomas L. Black
National Meteorological Center
Washington, DC 20233, U.S.A.

Abstract. Issues of the numerical representation of mountains in atmospheric models are reviewed. An example is given illustrating the origin of the most frequently considered type of errors resulting from terrain-following coordinates. Finally, a forecast obtained with the model using a so-called step-mountain coordinate is compared with that obtained when the same model is run with a standard terrain-following (sigma) coordinate.

I. INTRODUCTION

Since its introduction by Phillips in late fifties the terrain-following vertical coordinate has been by far the most predominant in atmospheric modeling. While a number of its problems were recognized relatively early, the simplicity it offered for the representation of mountains seemed to be more than a sufficient compensation for the problems. Furthermore, for longer than a decade, various methods were being devised to minimize the errors. These techniques were addressing the finite-difference methods almost exclusively used at that time.

In the early eighties, however, it became increasingly doubtful that the errors associated with terrain-following coordinates could indeed be adequately minimized. It was shown that the accuracy of the calculation of the pressure gradient force over steep mountain slopes is likely to deteriorate rather than improve with an increase in vertical resolution. On the other hand, a different technique using step-like representation of mountains with approximately horizontal coordinate surfaces was proposed and appeared to be an attractive alternative. For a comprehensive review of these various techniques proposed up to the mid-eighties as well as a description of the step-mountain system the reader is referred to Mesinger and Janjić (1985).

Efforts aimed at reducing or eliminating the pressure gradient force errors continued (e.g., Zheng and Liou 1986, Carroll et al. 1987). At the same time, new examples of potentially large errors were reported with regard to the terrain- following coordinate scheme in spite of a sophisicated design (Mesinger and Janjić 1987). A comprehensive step-mountain ("eta") coordinate model was developed and in extensive tests has been shown to be competitive with an operational state-of-the-art terrain following coordinate model. While evidence was presented indicating that the numerics of the model were primarily responsible for the improved results (Mesinger et al. 1990), no assessment was made of the extent to which the improvement came from the vertical coordinate as opposed to other numerical features of the model.

A recent study by Janjić (1989) shows that in spectral models, now dominant in global weather and climate simulations, pressure gradient force errors might be still greater than in finite-difference models. Only the terrain-following formulation appears to be practical for spectral models at present however.

II. AN ASYMPTOTIC LIMIT OF THE ERROR

An example which lends itself to analytic treatment and is useful for assessing errors of various schemes is that of a resting hydrostatic atmosphere in which the pressure gradient force must be zero. Given a numerical scheme, an assumed temperature profile, and the mountain slope, the pressure gradient force can typically be calculated in a straightforward way which hopefully indicates the magnitude of the error.

A recent calculation of this type for three schemes and three temperature profiles has been reported by Mesinger and Janjić (1987). Of the three schemes, the latest is the "θ-conserving" scheme of Arakawa and Suarez (1983) expanded to include horizontal differencing. Rather large errors were obtained, at some of the levels and temperature profiles, for all three of the schemes. In particular, a convergence problem was identified in all three of the schemes in the sense that no tendency was visible for the general magnitude of the error to decrease with increasing vertical resolution.

The three schemes considered are identical for the case of an isentropic atmosphere where the potential temperature $\theta = \Theta =$ const. Considering the case of zero pressure at the top of the model atmosphere ($p_T = 0$) the asymptotic value of the error as thicknesses of the layers $\Delta\sigma$ tend to zero is

$$-(\Delta_x \Phi_p)_k \to \frac{R\Theta}{p_0{}^\kappa}\left[\frac{\Delta_x p_S{}^\kappa}{\kappa} - \frac{\overline{p_S{}^\kappa}^x}{\overline{p_S}^x}\Delta_x p_S\right]\sigma^\kappa. \qquad (1)$$

Here Δ_x and $\overline{}^x$ are the centered two-point difference and averaging operators, respectively, applied along the direction of the x axis; Φ is geopotential; k is the vertical index, increasing downward, identifying quantities defined for the sigma layers; R is the gas constant; p_0 is the reference pressure used to define the potential temperature; κ is R/c_p, where c_p is the specific heat at constant pressure, and p_S is the surface pressure Furthermore, denoting the partial derivative and its centered two-point difference analog by ∂_x and δ_x, respectively, note that in the limit as Δx also approaches zero

$$\frac{1}{\kappa}\delta_x p_S{}^\kappa \to p_S{}^{\kappa-1}\partial_x p_S = \frac{\overline{p_S{}^\kappa}^x}{\overline{p_S}^x}\partial_x p_S.$$

Thus, in that limit, there will be no pressure gradient force error. However, for finite Δx, the right hand side of (1) will be different from zero, in spite of the exact finite-difference hydrostatic equation of the three schemes in the considered isentropic atmosphere case.

Attempting to reduce the problem by increasing the horizontal resolution actually tends to worsen the situation since the accompanying incorporation of more realistic mountains simply introduces more sloping terrain. Thus, representation of mountains seems to be an issue of a steadily increasing pı iority to atmospheric modelers.

III. A FORECAST EXAMPLE

It is not obvious to what extent pressure gradient force errors of a specific idealized example are relevant in actual atmospheric simulations. In addition, other problems are associated with terrain following coordinates. A potentially serious problem is that of horizontal advection with sloping coordinate surfaces such that the vertical velocity relative to these surfaces is required to compensate for their slope.

A study of model performance in actual weather situations thus seems to be the ultimate answer. It is our intention to complete such a study, an example will be shown here. A comprehensive (eta) prediction model was employed which can be used either with a terrain-following or with the step-mountain coordinate, the code and the schemes being the same. The case to be shown was the first case we looked at from the point of view of sensitivity to the vertical coordinate after an extended period of model development. It involves a major cold air outbreak along the eastern slopes of the Rockies.

The U.S. National Meteorological Center surface analysis for 1200 UTC 2 February 1989 is shown in the upper panel of Fig. 1. The 36-h sea level pressure forecast valid at the same time and obtained with the model using the terrain-following ("sigma") formulation is shown in the middle panel. Finally, the 36-h forecast valid at the same time obtained using the step-mountain (eta) coordinate is shown in the lower panel.

One feature favoring the eta result is the reduced noisiness of the eta integration with 21 centers printed as compared to 27 centers of the sigma map Note that these maps are printed without the usual smoothing prior to output so that unrealistic centers are obtained as the result of model noise. Another feature is the more accurate simulation of the southward extention of the cold air east of Rockies. For example, note that the sea level pressure over North Dakota in the eta integration is about 4 mb higher than it is in the sigma integration. Still higher sea level pressures are seen in the analyzed map. We intend to analyze such differences further and to report on our results more extensively at a later occasion

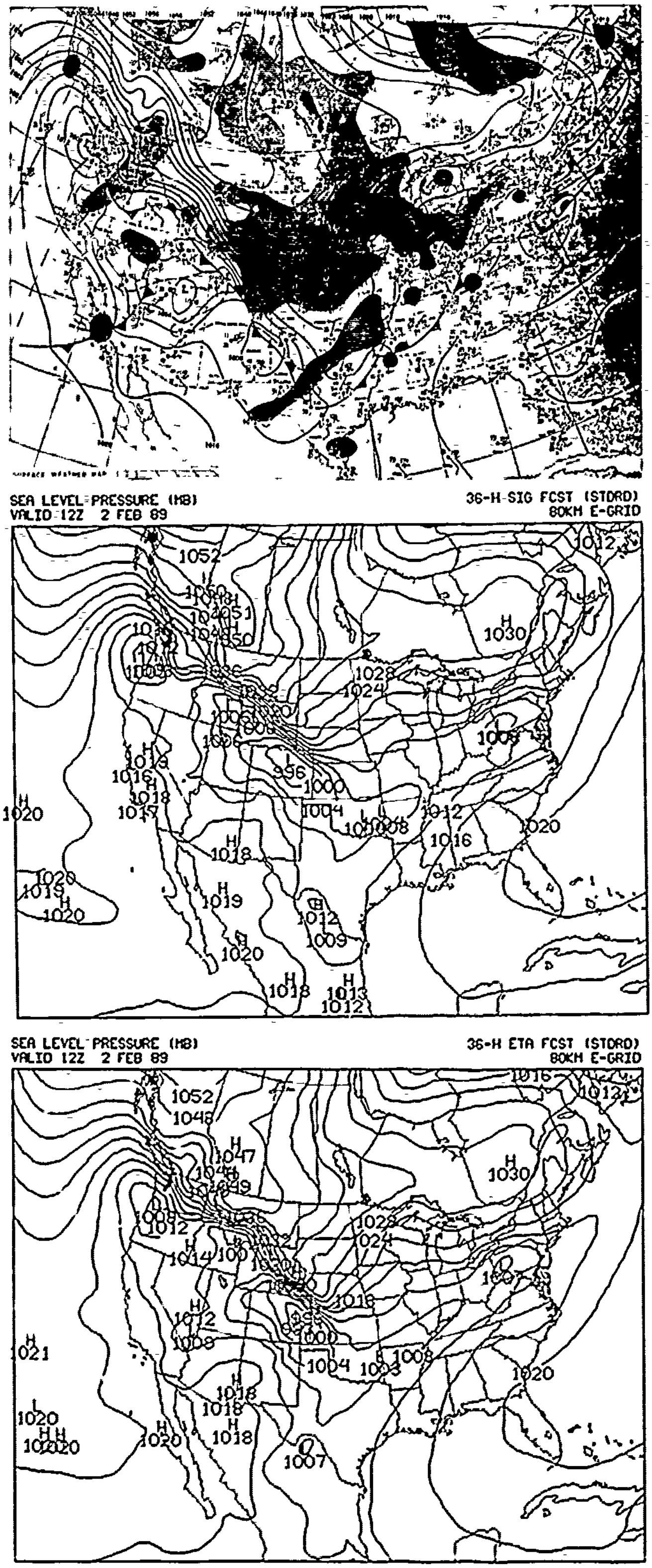


Figure 1. Analyzed (upper panel) and 36-h forecast (sigma system, middle panel, step-mountain system, lower panel) sea level pressure (mb) at 1200 UTC 2 February 1989

REFERENCES

Arakawa, A., and M. J. Suarez, 1983, Mon. Wea. Rev., **111**, 34-45.

Carroll, J. J., L. R Mendez Nunez, and S. Tanrikulu, 1987 Boundary-Layer Meteor., **41**, 149-169.

Janjić, Z. I., 1989, Mon. Wea. Rev., **117**, 2285 2292.

Mesinger, F., T. L. Black, D. W. Plummer and J. H. Ward, 1990, Wea. Forecasting, *5*, 483-493.

Mesinger, F., and Z. I. Janjić, 1985, Lect. Appl. Math., Vol. 22, Amer. Math. Soc., 81-120.

Mesinger, F., and Z. I. Janjić, 1987, Seminar/workshop 1986, Vol. 2, ECMWF, Shinfield Park, Reading, U.K., 29-80.

Zheng, Q., and K.-N. Liou, 1986, J. Atmos. Sci., **43**, 1340-1354.

NUMERICAL METHODS FOR A UNIFIED FORECAST/CLIMATE MODEL

M.J.P.CULLEN
Meteorological Office,
London Road,
BRACKNELL, Berks. RG12 2SZ, U.K.

Abstract Finite difference methods which combine the efficiency required for forecast models with the conservation properties required for long-term climate integrations are described. Methods of using non-oscillatory advection schemes within this framework are discussed.

I. INTRODUCTION

Numerical methods for forecast models are usually selected on the basis of efficiency and accuracy. Finite difference methods within this category are the semi-implicit and split-explicit methods, with further improvements available from using semi-Lagrangian advection. In climate change experiments global conservation properties must be enforced, and measures taken to ensure satisfactory long-term behaviour. Correct treatment of the energetics is one necessary requirement.

This paper extends the split-explicit scheme of Gadd (1978) to meet the climate modelling requirement. In order to enforce conservation and balance the energy conversion terms it is necessary to advect all the fields with a three-dimensional velocity field consistent with the continuity equation. In the original split-explicit method this was not satisfied because the vertical advection is carried out in the short timesteps and the horizontal advection in the long timestep. The requirements can be satisfied by instead including the advection of a basic state potential temperature in the short timestep, and all the rest of the advection in a long timestep, using as the advecting velocity the average mass-weighted velocity from the short timesteps. This makes the structure of a scheme similar to a standard semi-implicit scheme. In particular, the basic state potential temperature must be chosen according to the criteria established in Simmons, Hoskins and Burridge (1978).

Practical implementation of the scheme in a global model requires choices of numerical filtering and smoothing methods consistent with the requirements. Fourier filtering in high latitudes is applied to mass-weighted velocity fields, and to mass-weighted potential temperature and moisture increments, so that conservation properties are retained and the fields are not distorted. Numerical noise is removed with conservative high order filters. The order used depends on the resolution of the model.

There has been much recent interest in the use of advection schemes which prevent the development of spurious oscillations, e.g. Williamson and Rasch (1988). The most suitable way of constructing such schemes within the split-explicit framework is that introduced by Roe (1983), where advective increments calculated on the boundaries of contr l volumes are distributed upwind or downwind according to a suitable limiting criterion. The implementation of one of these is described.

II. A CONSERVATIVE SPLIT EXPICIT SCHEME

For the purposes of this paper, the scheme is written in Cartesian coordinates. The extension to spherical geometry is straightforward. The equations required to treat moisture are omitted, as are additional small terms included in the model to allow more accurate treatment of planetary scale flows. However, the details depend critically on the vertical coordinate, and are therefore written out for the hybrid coordinate actually used in the model. The coordinate is a mixture of pressure and normalised pressure as described by Simmons and Burridge (1981).

Start with data at time t for velocity components u and v, surface pressure p, and potential temperature θ. Then a series of short timesteps δt is taken, solving first

$$u^{t+\delta t}=u^{t} + \delta t\Big[\tfrac{1}{2}f(v^{t+\delta t}+ v^{t}) -$$

$$\overline{{}_{x}\frac{1}{\Delta x}\Big\{ \delta_x\phi + \frac{c_p\theta}{(\kappa+1)}\,\delta_x\Big[\frac{\Pi_{k+\frac{1}{2}}\,p_{k+\frac{1}{2}} - \Pi_{k-\frac{1}{2}}\,p_{k-\frac{1}{2}}}{\Delta p_k}\Big]}^{\,x}$$

at each level k, with a similar equation for v. f is the Coriolis parameter, c_p the specific heat of air at constant pressure, $\kappa = R/c_p$ where R is the gas constant, ϕ is the geopotential, and Π is the Exner function $(p/p_0)^{\kappa}$ with p_0 a reference pressure. The standard finite difference averaging notation is used.

The hydrostatic equation is approximated by

$$\phi_k = \phi_* - \sum_{m=1}^{k-1} c_p\theta_m(\Pi_{m+1/2} - \Pi_{m-1/2}) +$$

$$c_p\theta_k\Big[\,\Pi_{k-1/2} + \frac{(\Pi_{k-1/2}p_{k-1/2} - \Pi_{k+1/2}p_{k+1/2})}{(\kappa+1)\Delta p_k}$$

The special form of the last term is chosen to ensure angular momentum conservation.

The second half of the forward-backward step solves

$$p_*^{t+\delta t} = p_*^{t} + \frac{\delta t}{a^2}\sum_{m=1}^{TOP} D_m^{t+\delta t}$$

$$\theta^{t+\delta t} = \theta^{t} - {}^{1}\!/_{2}\Big[\,(\dot\eta\frac{\partial p}{\partial\eta})^{t+\delta t}_{k+1/2}(\theta_{Rk+1}-\theta_{Rk})^{t} +$$

$$(\dot\eta\frac{\partial p}{\partial\eta})^{t+\delta t}_{k-1/2}(\theta_{Rk} - \theta_{Rk-1})\,\Big]$$

where $\theta_R(\eta)$ is a basic state profile of θ calculated from an isothermal basic state with temperature 300°K and surface pressure 100000 pa. as used in standard semi-implicit models The theory behind the choice is exactly that in Simmons, Hoskins, and Burridge (1978).

Typically, three short steps are performed, follwed by a long advection timestep. The length of this step is written as Δt. The Heun two-step advection scheme is used. A second order version is written out, but a fourth order version is actually used for some of the applications of the model. The same order of accuracy must be used in both steps of the Heun scheme. The key to conservation is that the advection must be with the mass-weighted velocity field averaged over the three adjustment steps. Define

$$(U_k, V_k) = (\overline{u_k \Delta p_k}^{xy}, \overline{v_k \Delta p_k}^{xy} \cos \phi)$$

$$E_{k+1/2} = \left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)_{k+1/2}$$

as saved from the short timesteps. The finite difference equations for the first advection step are then

$$\Delta p_k^t \theta_k'^{\#} = \Delta p_k^t \theta_k'^{t} - \frac{\Delta t}{a \cos \phi} \left[\overline{\overline{U_k}^y \delta_x \theta_k'}^x + \overline{\overline{V_k}^x \delta_y \theta_k'}^y \right] -$$
$$\Delta t/_2 \left[E_{k+1/2}(\theta_{k+1}' - \theta_k') + E_{k-1/2}(\theta_k' - \theta_{k-1}') \right]$$

Similar equations, with the different spatial averaging appropriate for the position of the variables on the B grid, are used to advect the other variables. The second step is written

$$\Delta p_k^{t+\delta t} \theta_k'^{t+\delta t} = \Delta p_k^{t+\delta t} \theta_k' -$$
$$\frac{1}{2}\Delta t \left[(\Delta p_k^{t+\Delta t}/\Delta p_k^t) \underline{U}.\nabla \theta_k'^{t} + \underline{U}.\nabla \theta_k'^{\#} \right]$$

This form is chosen to ensure conservation under time differencing.

III. OSCILLATION-FREE ADVECTION SCHEMES

The most natural way of incorporating oscillation free advection schemes into this structure, while retaining conservation is that of Roe (1983). The standard advection scheme set out above can be considered as two stages. First calculate advective increments of the form

$$\overline{U_k}^y \delta_x \theta_k$$

at grid box boundaries. The second step is to apply these increments to adjacent grid points. This is done in the standard scheme by applying half the increment to the point on either side of the boundary. Upwind schemes can be generated by applying the whole increment to the point downwind of the boundary. Accurate oscillation free schemes are obtained by using a more accurate redistribution algorithm, several of which are set out in Roe's paper. The resulting schemes are algebraically equivalent to those obtained by starting from the conservation law form of the equations and applying a flux limiter. However, the use of short timesteps for the continuity equation and long timesteps for the advection makes it harder to set out the schemes in that form.

The schemes can then be applied to a three dimensional model on the sphere by combining three one-dimensional schemes. Thus the redistribution is calculated separately for each coordinate direction. Fourier filtering destroys the oscillation-free property. At high latitudes, the east west sweep must be repeated several times to avoid compromising the model timestep.

REFERENCES

Gadd, A.J. 1978
A split explicit scheme for numerical weather prediction. Quart. J. Roy. Meteor.Soc., 104, 569-582.

Roe, P.L. 1983
Some contributions to the modelling of discontinuous flows. AMS Lect. Appl. Math. 22, pt.2, 163-194.

Simmons, A.J. and Burridge, D.M. 1981
An energy and angular momentum conserving finite diffeence scheme and hybrid coordinates.
Mon. Weath. Rev., 109, 467-478.

Simmons, A.J., Hoskins, B.J. and Burridge, D.M. 1978
Stability of the semi-implicit method of time integration. Mon. Weath. Rev., 106, 405-412.

Williamson, D.L. and Rasch, P.J. 1988
Two-dimensional semi-Lagrangian transport with shape preserving interpolation. Mon. Weath. Rev., 117, 102-129.

COMPUTATIONAL DISPERSION PROPERTIES OF VERTICAL GRIDS FOR ATMOSPHERIC MODELS

Michael S. Fox-Rabinovitz
Laboratory for Atmospheres
NASA/Goddard Space Flight Center
Greenbelt, MD 20771, USA

ABSTRACT

Computational dispersion properties of different vertical grids, namely for regular, Lorenz, Charney-Phillips, and a new class of time-staggered grids have been studied using a linear baroclinic hydrostatic atmospheric model and intercompared in terms of frequency, phase and group velocity characteristics. It is shown that the widely used Lorenz grid, and also the Charney-Phillips grid as well as the new time-staggered versions of these grids have dispersion properties corresponding to a regular grid with twice the vertical resolution. They are computationally efficient due to enhanced effective vertical resolution. The scale ranges for which group velocities have the wrong sign are pointed out. The influence of higher (4th) order vertical approximation has been investigated and found to be relatively insignificant. The practical applicability of time-staggered vertical grids has been tested and proven effective within full time-space staggered grids for an experimental PE baroclinic model.

I. INTRODUCTION

The most widely used vertical grid is the Lorenz (1960) type staggered (or L-grid) which carries horizontal velocities and temperatures at the same levels, and vertical velocities at the intermediate levels. The advantage of the Lorenz grid is in its easy maintenance of conservation laws. The Charney-Phillips (1953) type staggered grid, or CP-grid carries vertical velocities and temperatures at the same levels and horizontal velocities at the intermediate levels. The advantage of this grid is its easy maintenance of integral constraints for a quasi-geostrophic flow.

New time-staggered versions of these vertically staggered grids, namely the time staggered L grid or the LTS grid, and time staggered CP grid or the CPTS grid are introduced. All staggered grids are computationally efficient due to enhanced effective vertical resolution compared to that of a regular (unstaggered) grid. Their dispersion properties are intercompared both with each other and against that of an analytical case in a manner similar to that used by Mesinger and Arakawa (1976) and Fox-Rabinovitz (1991) for horizontal grids.

II. THE DIFFERENTIAL CASE

Let us consider a linear baroclinic PE hydrostatic atmospheric model in $\zeta = \ell np$ coordinate system.

$$\frac{\partial u}{\partial t}+\frac{\partial \phi}{\partial x}-fv=0\ ,\ \frac{\partial v}{\partial t}+\frac{\partial \phi}{\partial y}-fu=0\ ,$$

$$\frac{\partial^2 \phi}{\partial t \partial \zeta}+c^2\Omega=0\ ,\ \frac{\partial u}{\partial x}+\frac{\partial v}{\partial x}+\frac{\partial \Omega}{\partial \zeta}=0, \tag{1}$$

where $u,v,\Omega=\frac{d\zeta}{dt}$ are velocity components, c^2 = const., ϕ geopotential, f constant Coriolis parameter. The solution has the form:

$$F=\overline{F}\,.\exp\left[i\left(kx+my+r\zeta-\nu t\right)\right], \tag{2}$$

where k, m, r are wave numbers, ν frequency. The dispersion relationship and horizontal and vertical group velocity components (HGVC and VGVC) are as follows (for k = m):

$$\nu=B\ ;\ HGVC=\frac{\partial \nu}{\partial k}=2B^{-1}c^2Kr^{-2}>0;$$

$$VGVC=\frac{\partial \nu}{\partial r}=-2B^{-1}c^2k^2r^{-3}<0; \tag{3}$$

where $B\equiv\left(f^2+2c^2k^2r^{-2}\right)^{1/2}$

These characteristics are used for comparisons of different grid dispersion properties. Note that the most important characteristic is the appropriate sign of group velocity components, namely HGVC should be positive and VGVC negative for all resolved scales.

III. VERTICALLY AND TIME-VERTICALLY STAGGERED GRIDS

The same dispersion characteristics are obtained for the system (1) approximated with a central difference scheme (CDS) for different vertically and time-vertically staggered grids, and compared with each other, and against the differential case (3). For simplicity a regular, or Arakawa A grid is used for horizontal approximations. The vertical grids considered are presented in Figs. 1-5.

We considered first what may be achieved by introducing 4th order CDS in the vertical instead of the 2nd order CDS. The sign of both VGVC and HGVC are wrong for both schemes in certain scale ranges, although for the 4th order CDS they are slightly smaller. Namely, for the 2nd order CDS the aforementioned signs are wrong for scales smaller than $L < 4\Delta\zeta$, whereas for the 4th order CDS the signs are wrong for scales $L < \sim 3.5\ \Delta\zeta$ which is quite comparable to the 2nd orders result. Therefore, the dispersion properties cannot be significantly improved by using the higher (4th) order approximation in the vertical with a regular (unstaggered) grid. The most significant feature of all vertically and time-vertically staggered grids considered is that they have definitely better dispersion characteristics than that of a regular (unstaggered) vertical grid. Both HGVC and VGVC have an appropriate sign for all resolvable scales $L < 2\Delta\zeta$ due to the higher effective vertical resolution. In general all staggered grids have dispersion properties which correspond to or are very close to those of a regular grid with twice the vertical resolution.

Note that semi-implicit or economical explicit schemes may be complemented with time-vertically staggered grids and the dispersion properties for the case are identical with those for a regular grid with doubled resolution. The examples of VGVCs for different vertical grids are presented in Fig. 6.

Note also that the upper and lower levels for time-vertically staggered grids may be the same for the adjacent time steps which is convenient for orography incorporation and upper boundary condition.

IV. EXPERIMENTS WITH TIME-VERTICALLY STAGGERED GRIDS

The time-vertically staggered grids have been combined with time-horizontally staggered grids within a full 4-D staggered grid approach. The experimental PE baroclinic model with the grid is tested and found to be computationally efficient. The results of regional and storm forecasts are encouraging.

REFERENCES

Charney, J. G., and N. A. Phillips, 1953: Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows. J. Meteor., 10, 17-99.

Fox-Rabinovitz, M. S., 1991: Dispersion properties of horizontal staggered grids for atmospheric and ocean models. To appear in Mon. Wea. Rev. in June.

Lorenz, E. N., 1960: Energy and numerical weather prediction. Tellus, 12, 364-373.

Mesinger, F., and Arakawa, 1976: Numerical methods used in atmospheric models, V. I. WMO/ICSU JOC. GARP Publ. Series, No. 17, 64 pp.

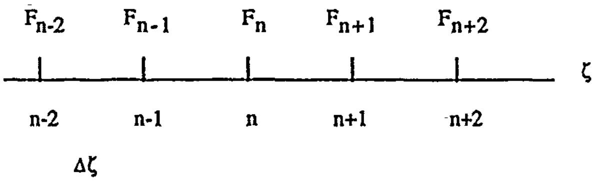


Fig. 1. A regular (unstaggered) vertical grid. F stands for model variables.

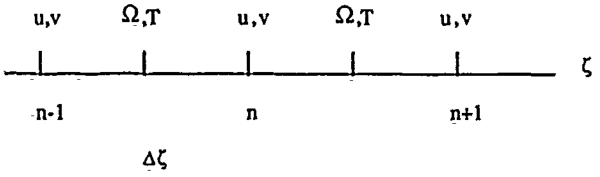


Fig. 2. The vertically staggered Charney-Phillips grid (1953), or CP grid.

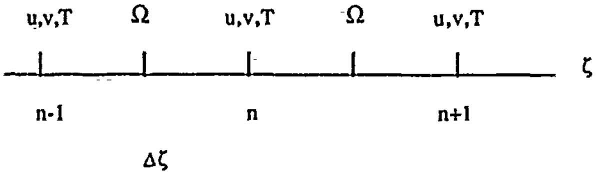


Fig. 3. The vertically staggered Lorenz grid (1960), or L grid.

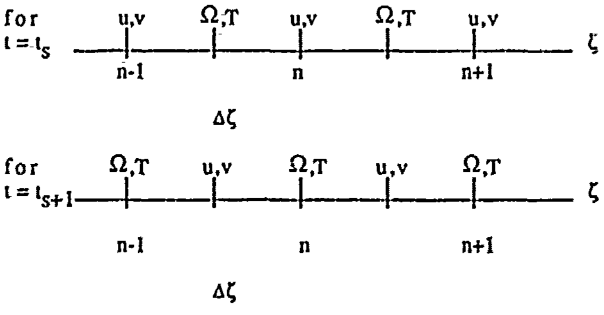


Fig. 4. The time-vertically staggered CP grid, or CPTS grid.

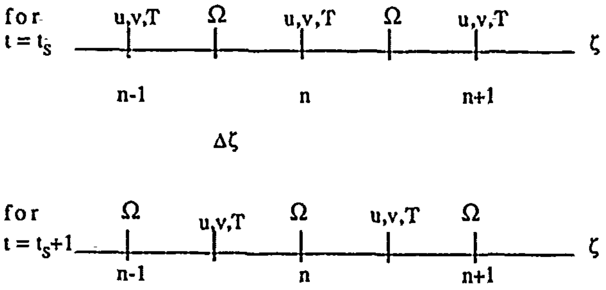


Fig. 5. The time-vertically staggered L grid, or LTS grid.

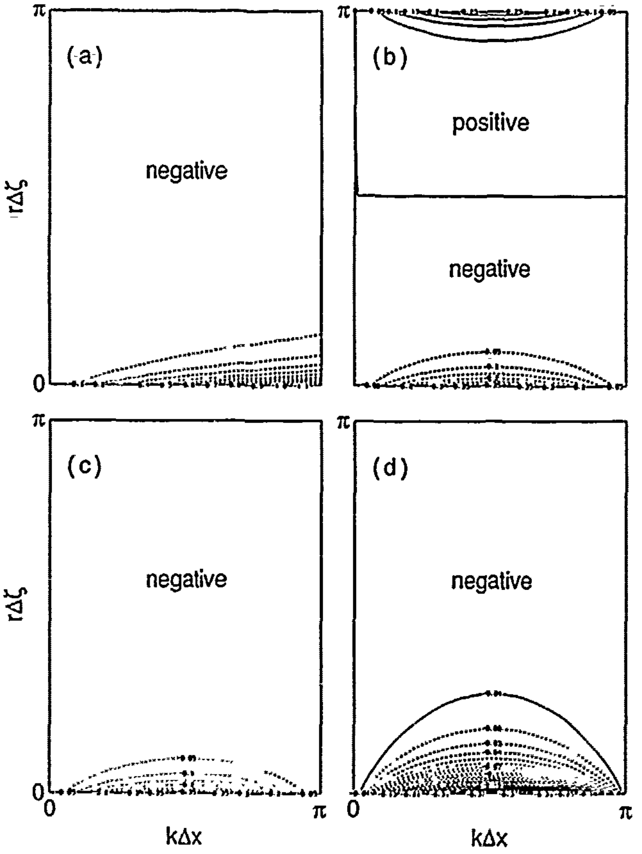


Fig. 6. Vertical group velocity components for the differential case (a); a regular vertical grid with the 2nd order CDS (b); vertically staggered L and CP grids (c); and time-vertically staggered LTS and CPTS grids (d).

ERRORS ASSOCIATED WITH HORIZONTAL TRUNCATION IN GLOBAL ATMOSPHERIC MODELS

FERDINAND BAER
Department of Meteorology
University of Maryland
College Park, MD 20742 USA

and

JIANJUN ZHANG
Department of Meteorology
University of Maryland
College Park, MD 20742 USA

I. INTRODUCTION

One aspect of weather and/or climate prediction which limits our capabilities is associated with computational errors arising from the numerical representation of the relevant differential equations describing the prediction system. Of the representational methods currently favored, global-function expansion (spectral), finite differencing in space or finite element functions, each has its own limitations, but the interrelationship amongst them has not been carefully discussed. Issues such as aliasing errors and computing speed compete when a representational choice is to be made. Whereas the spectral method is preferred for global calculations, it is not suitable for regional calculations with complex boundary conditions.

It is the intent of this report to present a procedure which, when applied in the spatial domain, could yield results which are equivalent to those derived from the spectral domain. If the equivalence can be demonstrated in application to general models, the choice of numerical representation would be one of preference or convenience rather than one associated with forecast quality. We shall lean heavily on linear theory to present our case.

II. PROCEDURE

We choose for our analysis the model developed by Kalney, et. al. (1977). This model is representative of the models in current use as prediction systems and is furthermore available in 4th order difference form at GSFC/GLA, if and when we need a comprehensive model to test our results. Since the model is global, it would also be possible to compare a spectral version with the existing finite-difference form. To understand the structure of the prediction equations, we linearize the model without forcing about a state of rest.

The resulting system yields to a separation in variables such that the horizontal dependence may be solved for independently of the vertical dependence, the separation constants are derived from the solution to the equations in the vertical dimension and are denoted as *equivalent depths* with corresponding eigenvectors which have been carefully studied by Baer and Ji (1989). If Fourier series are substituted for the longitudinal dependence, the shallow water equations, dependent only on latitude, result and have analytic solutions. These solutions are known as Hough functions and depend on the equivalent depth and the planetary wave number. There are an unlimited number of these functions which satisfy the equations, and a finite set would reflect modal truncation.

To understand how a finite-difference or finite element representation of the shallow water equations relates to the spectral (Hough) solutions, we have evaluated the latitudinally dependent equations on an equally spaced grid of points in latitude, and using both the 4th order differencing scheme and a second order finite-element scheme applied to the general model, we have developed two discrete systems, each of which yield a set of eigenvectors and eigenvalues, their number and character depending on the number of grid points selected. Since the true solutions are known (Hough functions), these eigenvectors can be compared to the Hough solutions and identified as true solutions (those which compare) or as false solutions (those which do not compare and consequently represent computational errors). If a one-to-one comparison exists and only the true eigenmodes are used to solve the linear system on the grid, then the results of the discrete representation would yield the exact solutions.

Should one consider a different model, or even the same one which was not valid globally, the exact solutions would not be apparent. Thus an alternate procedure is needed to distinguish the true modes from the false ones. We have done this by introducing the *shooting* method. Simply described, this method utilizes the eigenvalue for a given numerical eigensolution of the discrete equations, starts at one boundary and integrates the equation point-by-point to the other boundary. If the other boundary condition is thereby met, the eigenvalue and its associated vector satisfy the equation. If the boundary condition is not met, an adjustment to the eigenvalue is made and the process is repeated. If the solution converges to the boundary condition with minor adjustments of the eigenvalue, the numerical solution is probably true. If it diverges, the solution is false, i.e., no such vector satisfies the original equation. We compared the true and false solutions so identified to the Hough modes, which are by definition the true modes. By this procedure, we can identify the equivalence of a discrete model to a spectral model.

Finally, we have projected observational data fields onto a set of Hough functions and on the corresponding sets of eigenvectors derived from the discrete systems. Since the projections are done independently, some measure of amplitude loss which comes from discarding false modes is made apparent and is available to assess the applicability of the procedure.

III. EXPERIMENTS

The model equations discussed above, essentially the shallow water equations, were transformed to discrete systems by assuming (a) 4th order differencing over an equally spaced latitudinal grid and (b) second order finite element representation on the same grid. In both cases, the grids used were 5° latitude and 2.5° latitude. The corresponding longitude representation was for a continuous Fourier series and a fourth order differencing with the appropriate grid increment to allow for a maximum of 20 planetary waves.

The equations representing each differencing scheme were solved as a matrix problem and yielded eigenvectors and eigenvalues, their number appropriate to the number of grid points used. For example, if the grid increment was 5°, 105 vectors evolved. These vector sets were calculated for each planetary wave, $1 \leq m \leq 20$, and for the various equivalent depths. Assuming nine vertical levels, we evaluated for each of the nine equivalent depths. The minimum number of Hough modes appropriate to a given truncation ($\Delta\phi$) was established by identifying that set of Hough modes which were included amongst the eigenvector set of the discrete system, no more or no less.

The shooting method was applied to each set of eigenvectors of the discrete representation for each choice of parameters and for each numerical technique. Thus a set of true and false modes was determined for each experiment and was available for comparison with the associated Hough mode solutions.

Finally, several data fields were projected onto the eigenvectors derived from the various experiments, as well as onto the corresponding Hough modes, and the relative amplitude of the data in the true modes as contrasted to the false modes was assessed.

IV. RESULTS

The experiments outlined above have resulted in the following observations.

(a) Sensitivity to truncation in longitude results in variations observed only on the shortest planetary scales, which represent a $2\Delta\lambda$ increment. This confirms many previous observations.

(b) The distribution of modes between Rossby and gravity evolving from the solutions to the discrete (truncated) systems conforms to the expectations for the true (Hough) solutions; i.e., two-thirds of the solutions are gravitational and one-third are Rossby. However, some of the Rossby modes calculated are false because they propagate to the east. Almost half of the Rossby modes calculated for the 4th order system fall into this category, whereas considerably fewer have this property for the finite-element system.

(c) The shooting method works well and as expected. The modes which are defined as true by the shooting method are also Hough modes whereas the false modes are not found amongst the Hough mode solutions.

(d) A direct relationship has been found between the finite-difference increment selected and spectral truncation. Based on item (c) above, this correspondence can be calculated from either the number of true modes found by shooting or the number of relevant Hough modes determined. The results are sensitive both to the planetary wave number and to the vertical mode. Moreover, the response of the Rossby and gravity modes differ. We have tested these results for both $\Delta\phi = 5°$ and $\Delta\phi = 2.5°$, and the results are consistent. Consider the experiment for $\Delta\phi = 5°$ with 4th order differencing. For the external vertical mode and the longest planetary wave ($m=1$), almost one-half of the Rossby modes are true and 45 percent of the gravity modes are true. This translates roughly to 16 Hough modes for each set (Rossby, gravity-east and gravity-west). However, for the longest planetary wave and the sixth internal mode, only twelve Rossby Hough modes are true and only *two* gravity modes are true. For the external mode at planetary wave twenty, twelve Rossby modes are true but only five gravity modes are true. The implications here are clearly that spectral truncation and the corresponding finite-difference truncation are highly scale dependent.

(e) The impact of false modes on the integration of a nonlinear system may be assessed by considering their involvement in the energy exchange process. We have investigated the impact by projection of several data fields onto all the modes of a discrete system. As an example, consider the rotational kinetic energy of the external mode expressed by the modes of the 4th order differencing system. If we compute the ratio of energy in the true modes to the energy in all modes, we note that for the Rossby modes, this ratio decays from near 95 percent for the largest planetary wave to near 80 percent at wave number twenty. By contrast the ratio is near 40 percent for the largest planetary wave for gravity waves, and gradually increases to near 50 percent by wave twenty. Since the energy in both groups decays rapidly with wave number, and furthermore since the Rossby energy is an order of magnitude larger, the total energy in this vertical mode is described to better than 90 percent by the true modes. This result is similarly noted for the larger internal vertical modes.

V. CONCLUSIONS

The results of our calculations indicate that a correspondence can be made between spectral and grid truncation, and based on our analysis with a relatively simple model, it is highly scale dependent. Whereas for the longest planetary waves and the external vertical mode the conventional concept that one wave for each two grid points applies, very few spectral components represent many grid points for short planetary waves and internal vertical modes.

Truncation decisions can be made by considering the distribution of observed variables in terms of the appropriate eigenvectors of the system to be predicted. Once a decision on the number of false modes which can be tolerated is made, the appropriate truncation can be chosen and a grid or spectral truncation can be selected. Aliasing errors during nonlinear integrations can be reduced by filtering the initial data of the unwanted false modes. Because those modes exist in the prediction system, their amplitude will grow in time due to nonlinear interactions To maintain control of these unwanted modes, they may be filtered periodically during the integration cycle. This procedure is clearly not needed in a spectral model.

Tests with a shallow water nonlinear model have been made and indicate how fast false modes grow with time during integration. Such calculations provide a time scale for filtering of the false modes and also give an indication of how successfully a filtered model can provide successful forecasts.

REFERENCES

Baer, F., and Ming Ji, 1989: Optimal vertical discretization for atmospheric models. *Mon. Wea. Rev.*, 117, 391-406.

Kalnay, E., et. al., 1977: The 4th order GISS model of the global atmosphere. *Beit. Phys. Atm.*, 50, 299-311.

POLYNOMIALS AS A SUBSTITUTE FOR LEGENDRE FUNCTIONS IN SPECTRAL MODELS

Isidore M. Halberstam
ST Systems Corporation
109 Massachusetts Ave.
Lexington, MA 02173 U.S.A.

Abstract - A method for expanding dependent variables in numerical models of the global atmosphere is discussed. The method is shown to be compatible with spectral methods and a procedure for converting spectral coefficients to polynomial coefficients is demonstrated. The polynomial method proves more efficient on a scalar computer and has potential to be more efficient than traditional spectral algorithms.

I. INTRODUCTION

During the development of spectral models of the atmosphere in the 1970s, most researchers realized that a major difficulty with spectral models is the evaluation of Legendre functions. Although swift and accurate methods exist for the computation of Legendre functions, it was not feasible to store all the necessary values of the functions and to access them frequently during numerical integration because of the drag IO normally imparts on computer processing.

Several researchers have, therefore, looked into the possibility of expressing the Legendre functions in terms of Fourier expansions divided by some power of $(1-\mu^2)^{1/2}$, i.e., $\cos\phi$ where $\mu = \sin\phi$, and ϕ is latitude. Merilees (1973) suggested that the Legendre function be expressed as $F_m(\mu)/(1-\mu^2)^{m/2}$, where F_m is an expansion of sin $k\phi$, where k ranges from 0 to m, m being the zonal wave number. Orszag (1974) suggested that the function be represented as $G_m(\mu)/(1-\mu^2)^{s/2}$ where $G_m(\mu)$ is also an expansion of sin $k\phi$ while s can be either 1 or 0 depending on whether m is even or odd. Yee (1980) extended the argument so that the Legendre function could be expressed completely as a Fourier series, where cosine terms are kept for even functions and sine terms are kept for odd functions.

In this study, we offer an alternative more compatible with Legendre functions and hence with spectral models. It involves shifting from expansions in terms of Legendre functions to expansions in polynomials of μ, while keeping the representation of the fields spectral.

II. SPECTRAL EXPANSIONS

To evaluate a given variable, A, which is a function of latitude, ϕ, and longitude, λ, from its spectral coefficients, A_n^m, we execute the sum

$$A(\phi,\lambda) = \sum_{m=-M}^{M} \sum_{n=|m|}^{|m|+N} A_n^m P_n^m(\phi)\, e^{im\lambda} \qquad (1)$$

where $P_n^m(\phi)$ are the associated Legendre functions. The Legendre functions themselves are defined as

$$P_n^m = (1-\mu^2)^{m/2} \sum_{j=0}^{n-m} a_j^{n,m}\mu^j.$$

The coefficients $a_j^{n,m}$ need be derived only once and stored for future use. Because P_n^m is even for n-m even and odd when n-m is odd, only of the order of (N-M) /4 coefficients need be saved for each m under rhomboidal truncation, where N represents the largest meridional wave number of the expression. There are various ways to derive the $a_j^{n,m}$, but there is no space here to expound upon them. Once they are known it is possible to substitute into (1) to derive the Fourier coefficients, $A_m(\phi)$ of $A(\phi,\lambda)$, given as

$$A_m(\phi)\ (1-\mu^2)^{-m/2} = \sum_{n=|m|}^{|m|+N} A_n^m \sum_{j=0}^{n} a_j^{n,m}\mu^j,$$

Rearranging terms gives

$$A_m(\phi)\ (1-\mu^2)^{-m/2} = \sum_{r=0}^{N} \sum_{p=r}^{N} A_{|m|+p}^m a_r^{|m|+p,m}\mu^r = \qquad (2)$$

$$\sum_{n=0}^{N} C_n^m \mu^n$$

where $C_r^m = \sum_{p=r}^{N} A_{|m|+p}^m a_r^{|m|+p,m}$.

Thus, given A_j^m and the coefficients $a_j^{n,m}$, we can determine the coefficients C_r^m and evaluate the variable in terms of powers of μ. The summation leading to C_r^m is a very swift process, especially on vector computers and, of course, the summation of powers of μ in (2) is much quicker than the evaluation of Legendre functions.

III. INVERSE TRANSFORMS

In the course of most spectral model processing, it is necessary to obtain the spectral coefficients from the values of the parameters, e.g., for non-linear terms this is normally performed at every time step. The polynomial approach can also furnish the same spectral coefficients but without explicitly calculating the Legendre functions and performing the related Gaussian integration. As with the spectral method, the function $A(\phi,\lambda)$ is decomposed by Fourier transform to obtain the Fourier coefficients at N+1 latitudes. We then divide $A_m(\phi)$ by $(1-\mu^2)^{m/2} = \cos^m\phi$, provided $\phi \neq \pm \pi/2$. We now solve the system of N+1 linear equations given by (2) to obtain the coefficients (C_j^m), j=0, ..., N. By invoking the relationship between the known C_j^m and $a_j^{n,m}$ and the unknown A_r^m, one can easily solve for the spectral coefficients recursively by

$$A^m_{|m|+N} = C^m_n / a_N^{|m|+N,m}$$

$$A^m_{|m|+N-1} = \frac{C^m_{N-1} - A^m_{|m|+N} a_{N-1}^{|m|+N,m}}{a_{N-1}^{|m|+N-1,m}}$$

$$\vdots$$

$$A^m_{|m|+N-k} = \frac{C^m_{n-k} - \sum_{r=N-k+1}^{N} A^m_{|m|+r} a_{N-k}^{|m|+r,m}}{a_{N-k}^{|m|+N-k,m}} \tag{3}$$

$$\vdots$$

$$A^m_{|m|} = \left(C^m_0 - \sum_{r=1}^{N} A^m_{|m|+r} a_0^{|m|+r,m}\right)\left(a_0^{|m|,m}\right)^{-1}$$

for each m. Inverting an upper triangular matrix would be the equivalent operation and may prove swifter.

IV. CONCLUSIONS

In experiments executed on a CDC Cyber 750 (an older, scalar machine), the expansion of temperature from spectral coefficients with rhomboidal truncation of 30 using the standard Legendre function approach over 60 latitudes took 1.472s of CPU time as opposed to .525s for the polynomial method. The inverse transform when performed by Legendre functions and Gaussian integration took 330s to evaluate the spectral coefficients for temperature, moisture, and velocity at 12 levels but only 84s with the polynomial method. Tests on faster vector processing machines have not been performed to date, but savings in time may be expected there as well. It must be remembered, as well, that the proposed method here is completely spectral and is therefore completely compatible with other spectral models.

There are some drawbacks to the method, however, which must enter in its evaluation. First, expressing the Laplacian is much simpler with Legendre functions than with polynomials, but the polynomial expression is still tenable. Second, it is necessary to store the coefficients of the powers of the sines that generate the Legendre functions, but this, again, may not prove too difficult especially with large mainframes. Third, accuracy near the poles may present a problem, but this affects Legendre functions as well. All in all, the polynomial approach may serve as a beneficial substitute for conventional spectral methods.

V. ACKNOWLEDGEMENT

Research and preparation of this document have been funded by the U.S. Air Force under Contract No. F19628-88-C-0072.

REFERENCES

Merilees, P. E., 1973: An alternative scheme for the summation of a series of spherical harmonics, J. Appl. Meteor., 12, 224-227.

Orszag, S. A., 1974: Fourier series on spheres, Mon. Wea. Rev., 102, 56-75.

Yee, S. Y. K., 1980: Studies on Fourier series on spheres, Mon. Wea. Rev., 108, 676-678,

A DUAL-RESOLUTION SEMI-IMPLICIT METHOD FOR ATMOSPHERIC MODELS

SAMUEL Y.K. YEE
Geophysics Laboratory (AFSC)
Hanscom AFB MA 01731-5000

I. Introduction

There are principally two types of waves represented in current operational atmospheric models: Rossby-waves and gravity waves. The former typically propagate at speeds of about 30 m/s and the latter at speeds up to 300 m/s. In the early days of numerical weather prediction (NWP), because of their diverged wavelengths and high speeds, gravity waves were treated with respect but also with awe. The common wisdom was to filter out the shorter ones and treat the longer ones gingerly. Thus, Robert (1969) proposed a so-called semi-implicit method in which the model terms responsible for the fast-moving gravity waves were treated by an unconditionally stable but computationally more costly implicit time-scheme, and the remaining terms were treated by a much simpler explicit time-scheme. This approach enabled us to use larger time-steps without sacrificing accuracy and was considered to be a computational milestone in NWP. Increasing emphasis in recent years on smaller-scale (the so-called mesoscale) dynamics and physics means that we must now solve model equations at higher resolutions. However, the current practice of discretization in which gravity waves are represented at the same spatial and temporal resolutions as Rossby-waves also means that, due to constraints in computing resources, we can have high resolution models only for pre-designated small geographical areas. Examples of such localized models are the USAF Global Weather Central's "Relocatable WindowModel" and the U.S. National Meteorological Center's "Nested-Grid Model."

We propose here a new approach in which different resolutions are adopted for different terms in the model equations. For example, since gravity waves typically not only propagate ten times faster than longer Rossby-waves, but also change much faster as functions of space and time, we may, therefore, discretize the model equations on a dual grid, both spatially and temporally, evaluate the terms responsible for the gravity waves on a finer grid, and evaluate the remaining terms on a coarser grid. The distinction between the local refinement method and this new approach, which we shall tentatively call the dual grid method, can be stated simply as follows: In the former, we start with a coarse grid and evaluate at increasingly finer resolutions for decreasingly smaller portions of the model domain. At a given level of refinement, all the terms in the model equations, however, are evaluated at the same resolution. In the latter, we evaluate for the entire domain each term in the model equations at a resolution appropriate for the scales of the phenomena it represents.

While it may not be immediately apparent, the underlying principles of this method are akin to those of the multigrid method (McCormick, 1989). shorter waves are treated on a finer grid, and longer ones on a coarser grid. In the example above, we have, however, taken advantage of prior information on the scales of the modeled waves.

II. A Shallow Water Model

To validate the dual grid concept outlined in Section I, we shall use a simple but illustrative "shallow" water model in which the wavelength of the shortest waves is much longer than the depth of the fluid so that the fluid is largely in hydrostatic equilibrium. The one-dimensional equations governing the velocities u and v and the depth h in a viscous "shallow" fluid on an f-plane can be written (Seitter, 1986)

$$(hu)_t = -(uhu)_x + f(hv) - g(h^2/2)_x + \nu(hu)_{xx} \qquad (1)$$

$$(hv)_t = -(uhv)_x - f(hu) + \nu(hv)_{xx} \qquad (2)$$

$$(h)_t = -(hu)_x \qquad (3)$$

where g is the acceleration of gravity, f is the Coriolis parameter, and ν is an eddy viscosity. The height-weighted velocities, hu and hv, are analogous to the pressure-weighted velocities used as prognostic variables in many mesoscale hydrostatic models and the continuity equation, (3), is analogous to the pressure tendency equation in a hydrostatic model.

We nondimensionalize the equations using the undisturbed height, H, as a scale. Thus, letting primes denote the nondimensional quantities we have

$$\begin{aligned} h &= H h' \\ u &= \surd(gH)\, u' \\ v &= \surd(gH)\, v' \\ t &= \surd(H/g)\, t' \\ x &= H x' \\ f &= \surd(g/H)\, f' \\ \nu &= H\surd(gH)\, K'. \end{aligned} \qquad (4)$$

Then the nondimensional equations are (dropping primes)

$$(hu)_t = -(uhu)_x + f(hv) - (h^2/2)_x + K(hu)_{xx} \qquad (5)$$

$$(hv)_t = -(uhv)_x - f(hu) + K(hv)_{xx} \qquad (6)$$

$$(h^2/2)_t = -(h-H)(hu)_x - H(hu)_x. \qquad (7)$$

Note that K may now be regarded as an inverse Reynolds number and that we have separated the height-weighted divergence term in (7) into perturbed and undisturbed parts. Following the choices of Seitter, we let K = 0.0078, f = 10^{-4} s^{-1}, and H = 8 km (approximately the density scale height). This yields an external gravity wave speed of $\surd(gH)$ = 280 m/s.

III. The Dual Grid Method

The two key issues in implementing the dual resolution method are: (a) the identification of the model forcing terms which contribute to shorter gravity waves and thus warrant higher resolutions, (b) the design of an intergrid information transfer procedure between the finer grid $d(\delta x,\delta t)$ and the coarser grid $D(\Delta x,\Delta t)$, where δ and Δ denote increments in grids d and D, respectively. The identification problem is not always easy. In our case, however, it is known that the imbalance between the pressure gradient and the linear divergence terms are largely responsible for the gravity waves. Although the role of the viscous terms is to simulate damping, we also discretize them at a higher resolution on the grounds that they represent smaller scale phenomena and that in more realistic models, K is a function of the flow field itself. For intergrid transfers, we simply use discrete Fourier smoothing and interpolation.

Thus given initial conditions on d, we

(1) smooth values of dependent variables on d to get smoothed values on D;
(2) compute longwave tendencies R(Δ) on D;
(3) interpolate R(Δ) to get R(δ) on d;
(4) compute gravity wave tendencies g(δ) on d;
(5) time-integrate on d, using R(δ) from (3) and g(δ) from (4) above;
(6) repeat steps (4) and (5) for (Δt/δt) times;
(7) repeat steps (1) through (6).

The main point here is that time integration is done on the finer grid at time intervals δt, but with the longwave tendencies computed only on the D grid. Another way of looking at it is that while longwave tendencies are computed less frequently on a coarser grid, they are interpolated onto the finer grid and then fed into the total tendencies at the smaller δt interval through step (5). It should be noted that step (1) ensures finer grid/coarser grid interactions and that this selective adaptation of grid resolutions according to the characteristic scales of the modeled phenomena should be done both spatially and temporally. Although our multi-resolution approach, in which the sizes of the time-steps are tailored to wavelengths and wave-speeds, enables us to use the computationally much simpler explicit time-scheme, for the studies reported here we have incorporated Robert's semi-implicit time scheme for pedagogical purposes.

IV Preliminary Results

The procedure given in Section III has been tested with the shallow water model. For compariscn purposes, model equations were solved in two different ways. In one, we simply used a centered-difference, explicit time scheme with a weak time filter (filtering coefficient α = 0.02) to couple even/odd time steps in the conventional manner. In the other, we solved a set of equations derived from (5), (6) and (7) using the dual-grid semi-implicit method. Periodic boundary conditions were used for a 38,400 km domain which contains 240 finer mesh points (δx = 160 km), but contains 15, 30, 60, and 240 mesh points, respectively, for various coarser grids. Time-step δt was set at 60 seconds, with the coarser grid Δt = 2δt. A small δt was used deliberately, although a 300 second δt has been used in some test cases for time-integrations up to 5 days.

Initially the fluid was assumed to be at rest. A sudden disturbance one-tenth the size of the domain was then imposed on h at the center of the domain, as shown schematically by the upper solid curve in Fig. 1. (Only half of the domain is shown.) The other curves in Fig. 1 show the height dis'ributions of the free surface at model times 4, 8, 16, and 24 hours. (We have displaced the curves vertically for clarity in the display.) We see that the free surface waves propagate both up and down stream with diminishing amplitudes during the first 24 hours in the model. In fact, the primary waves shown for hour-24 are retrograde waves originating at the center of the domain. In 24 hours, they have propagated slightly more than half of the domain. Fig. 2 shows the energy spectra of the height fields for the longest 25 waves at t = 0, and, at 24 hours as given by the two methods of solution. Initially, the energy spectrum is relatively smooth, with a single peak at wave numbers 15 and 16. At 24 hours, a bimodal distribution has developed in both model solutions, apparently due to the rapid removal of energy at the initial spectral peak. Furthermore, even in the cases shown (dual grid solution with Δx = 16δx), where the solutions diverge the most and where the energy spectra seem to be off-set by one or two wavenumbers, on the whole the two solutions possess remarkably similar spectral characteristics.

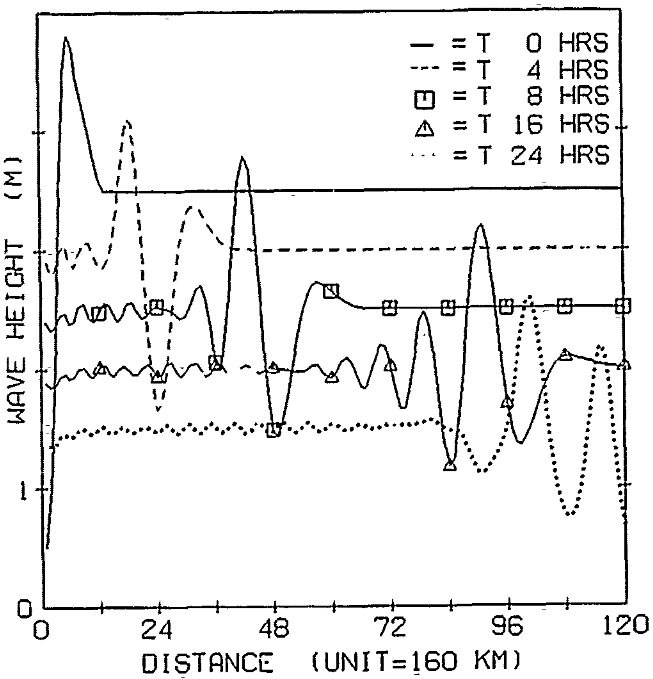


Fig. 1 Time-evolution of surface waves

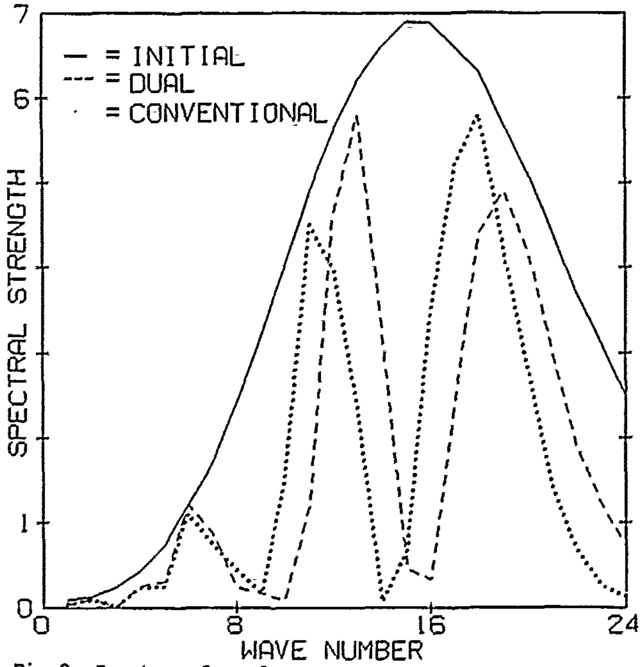


Fig 2 Spectra of surface waves at t = 0 and 24 hours

Acknowledgements: This research is partially supported by the Office of Scientific Research of the U.S. Air Force under Project 2304G1.

References

McCormick, S.F., 1989: Multilevel Adaptive Methods for Partial Differential Equations. Society for Industrial and Applied Mathematics, Philadelphia.

Robert, A.J., 1969: The Integration of a Spectral Model of the Atmosphere by the Implicit Method. Proceedings of the WMO/IUGG Symposium on Numerical Weather Prediction. Japan Meteorological Agency, Tokyo.

Seitter, K.L., 1986: The Specification of Lateral Boundary Conditions in Three-Dimensional Mesoscale Numerical Models. AFGL-TR-86-0005, AF Geophysics Laboratory, Hanscom AFB, Massachusetts.

Optimal Instability of Shear Flows in the Initial Value Problem

Enda O'Brien
University of Miami
Division of Meteorology and Physical Oceanography
4600 Rickenbacker Causeway
Miami, FL 33149, U.S.A.
Tel. (305) 361-4032

Abstract A variational principle is used to find those perturbation structures which have the fastest divergence from a basic state in linearized shear flows, and for finding the initial rate of divergence. The technique requires the existence of a positive definite quantity (e g , energy or potential enstrophy) whose growth rate can be expressed as a function of the instantaneous perturbation structure. The problem reduces to a fourth-order nonlinear ordinary differential equation for the structure function, which can be solved using various iterative numerical techniques. Green's model of baroclinic instability [1] is presented as an illustration of the method.

1 Introduction

A large class of waves in shear flows have instantaneous growth rates significantly larger than normal mode growth rates [2]. A consequence of this is that error growth in numerical forecast models can be faster than predicted by calculations of the first Lyapunov exponent (or growth rate of the most unstable normal mode). Questions which then arise are. is there a bound on the instability of these non-modal disturbances? If so, what is it and what is the structure which attains this bound?

In order to define a measure of instability, we seek a norm such that the perturbation growth rate in that norm can be expressed entirely as a function of the perturbation structure and the basic state flow. The perturbation structure can then be varied in order to maximize the growth rate.

2 Green's model

Consider a basic state wind $U(z)$, in a Boussinesq fluid with constant stratification on a β-plane, confined between rigid surfaces at $z = 0$ and $z = 1$. The governing equations are the linearized nondimensional quasi-geostrophic potential vorticity equation, along with the boundary conditions

$$\begin{aligned} q_t &= -Uq_x - vQ_y, \quad 0 < z < 1 & (1)\\ \psi_{zt} &= -U\psi_{zx} + vU_z, \quad z = 1 & (2)\\ \psi_{zt} &= -U\psi_{zx} + vU_z - r\nabla^2\psi, \quad z = 0. & (3)\end{aligned}$$

Here lowercase letters represent perturbation quantities, while uppercase letters refer to basic state quantities. The basic state potential vorticity gradient is Q_y, defined by:

$$Q_y = \beta - U_{zz}, \tag{4}$$

The dimensionless parameter r represents Ekman pumping at the ground. The perturbation streamfunction $\psi(x, y, z, t)$ is related to the wind components by $u = -\psi_y$, $v = \psi_x$, and to the potential vorticity q by

$$q = \nabla^2\psi + \psi_{zz}. \tag{5}$$

An energy equation is obtained by multiplying (1) by $-\psi$ and proceeding in the usual way. We obtain.

$$\frac{dE}{dt} = -\int_0^1 \overline{vq}U dz - r\overline{(\nabla\psi)^2}|_{(z=0)}. \tag{6}$$

Overbars represent a horizontal domain average. The total energy E is defined by:

$$E = 0.5\int_0^1 [\overline{(\nabla\psi)^2} + \overline{(\psi_z)^2}]dz. \tag{7}$$

The instantaneous energy growth rate σ_E is now defined as $E^{-1}dE/dt$.

An equation for the perturbation potential enstrophy H is obtained by multiplying (1) by q and proceeding as for the energy equation. The potential enstrophy growth rate σ_H is defined as $H^{-1}dH/dt$. For an arbitrary perturbation ψ there is no reason to expect σ_E and σ_H to be equal.

3 Constraints

Consider first the inviscid case (i.e., $r = 0$). Where $Q_y \neq 0$, we can multiply (1) by q, average horizontally, divide across by Q_y, and integrate vertically to obtain

$$\frac{d\mathcal{A}}{dt} = -\int_0^1 \overline{vq}\, dz \equiv 0, \tag{8}$$

where the wave action (or pseudomomentum) $\mathcal{A}$ is defined (formally) by

$$\mathcal{A} = 0.5\int_0^1 (\overline{q^2}/Q_y)dz. \tag{9}$$

Where Q_y vanishes the definition of $\mathcal{A}$ in (9) can be modified straightforwardly. Eq. (8) implies that $\mathcal{A}$ is a constant. We set $\mathcal{A} = 0$ since this is the only value for which $\mathcal{A}$ is independent of the perturbation amplitude, which is arbitrary.

When surface friction is included, the appropriate constraint becomes $\sigma_A = \sigma_X$, where σ_X is the growth rate in some norm (e.g., the energy or potential enstrophy norm), and the wave-action growth rate σ_A is defined as $\mathcal{A}^{-1}d\mathcal{A}/dt$.

4 The variational principle

The problem now is to maximize σ_E (or σ_H) subject to the constraints derived above. In the inviscid case, we render stationary the functional $\mathcal{L}$, where

$$\mathcal{L} = \sigma_E - \lambda\mathcal{A}. \tag{10}$$

In the case with friction, (10) is replaced by

$$\mathcal{L} = \sigma_E - \lambda(\sigma_E - \sigma_A). \tag{11}$$

For optimal potential enstrophy growth rates, σ_E is replaced by σ_H in (10) and (11). Now let $\psi(x, y, z, t) = \hat{\psi}(z, t)e^{i(kx+ly)} + cc.$ From here on the method is exactly analogous to the derivation of Lagrange's equations from a variational principle [3].

The same procedure is followed for all cases. For optimal energy growth in the inviscid case, we obtain a fourth order nonlinear ordinary differential equation as a structure equation for $\hat{\psi}(z)$ (henceforth dropping the "hat"). For $U_z = const$ (the classical Green's model) this equation reduces to:

$$\psi_{zzzz} - (2K^2+\gamma\sigma_E)\psi_{zz} + 2i\gamma kU_z\psi_z + (K^4+\gamma\sigma_E K^2)\psi = 0, \tag{12}$$

where $\gamma = Q_y/(\lambda E)$ and $K^2 = k^2 + l^2$.

Boundary conditions may be obtained for (12) in exactly the same way that the boundary conditions (2) and (3) are obtained for (1). This yields:

$$\psi_z = (ikU_z/\sigma_E)\psi, \quad z = 0;\ z = 1. \tag{13}$$

Since growth rate is independent of the wave's amplitude and phase, these quantities may be arbitrarily fixed at one level to provide a third condition on (12). A fourth condition is the integral constraint $\mathcal{A} = 0$, which is necessary to determine the value of the multiplier λ.

Assuming that a consistent value of σ_E exists, the general solution to (12) has the form

$$\psi = \sum_{i=1}^{4} A_i e^{m_i z}, \tag{14}$$

where m_i are the roots of a quartic equation and A_i are (complex) constants determined by the boundary conditions and constraints.

One approach to solving the problem (12) with its boundary and integral conditions is to fix λ, E and σ_E initially at some arbitrary values. The "constants" A_i in (14) can then be determined. This solution can be used to update σ_E and E in (12). Proceeding iteratively in this way, the values of σ_E and E used in (12) may or may not converge to values consistent with their definitions. Convergence only occurs for isolated values of λ

The approach outlined above was used in solving the Eady problem ($\beta = 0$). For the general problem, however, the optimal structure for ψ was obtained by discretizing in z, thereby reducing (12) to an algebraic problem which was solved using standard routines.

5 Results

In the results presented here, $\beta = 0.5$ and the optimization and eigenvalue problems are both solved in a 20-layer domain. Where friction is included, a value of $r = 2$ is used.

Fig. 1 shows σ_E, σ_H, and the growth rate for quadratic functions of the most unstable normal mode (referred to as σ_N) as functions of zonal wavenumber k, when σ_E is optimized. The normal mode growth rates (σ_N) show the well-known separation between the so-called Charney modes (for $k > 1.75$) and the Green modes ($k < 1.75$). The curves for optimal σ_E, and the corresponding σ_H all decrease monotonically as k increases and do not have a short-wave cutoff, at least in the inviscid case.

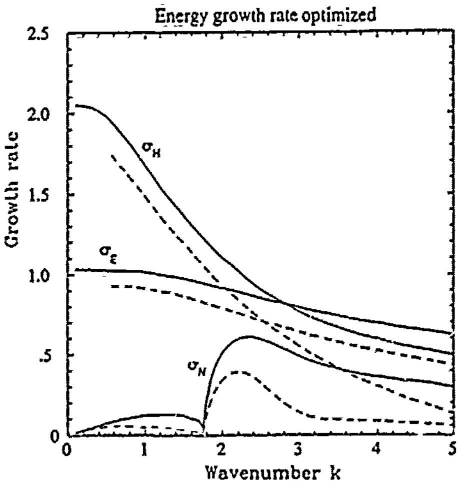


Fig. 1. Growth rate as a function of zonal wavenumber k ($l = 0$) for the most unstable normal modes (quadratic growth rate σ_N), the optimal energy growth rate (σ_E), and the corresponding potential enstrophy growth rate (σ_H). Solid curves are from the inviscid case, dashed curves from the case with friction.

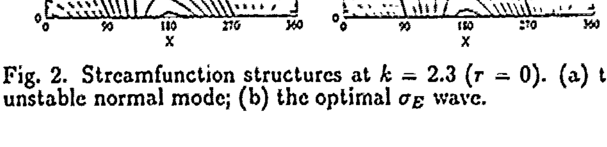


Fig. 2. Streamfunction structures at $k = 2.3$ ($r = 0$). (a) the unstable normal mode; (b) the optimal σ_E wave.

Fig. 2a shows the structure of the most unstable normal mode (at $k = 2.3$) in the inviscid case. The phase tilt is concentrated near the bottom of the domain. The structure which has optimal energy growth rate at the same wavenumber is shown in Fig. 2b, it has a phase tilt almost uniform with depth. This difference between normal modes and optimal structures seems to be quite general.

Fig. 3 shows the time-evolution of energy and potential enstrophy growth rates from an initial condition of optimal σ_E at $k = 2.3$, for both the inviscid and viscid cases. All cases show a similar, smooth evolution pattern, with the normal mode structures essentially established by $t = 10$. However, the waves have lost their character as optimal structures by $t = 2$.

Normal modes in the vertically discretized model have also been obtained using the variational principle by imposing the extra constraints that energy and potential enstrophy growth rates *in each layer* all be equal. The phase speed c also emerges from this analysis. An N-layer model produces a nonlinear algebraic system of $4N-2$ equations in $4N-2$ unknowns — clearly an inefficient way to find normal modes. However, the exercise serves to demonstrate the generality of the optimization theory, and in practise provides a check on our other results.

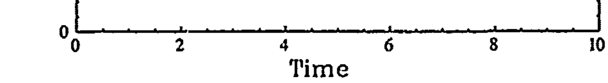


Figure 3. Time evolution of energy (solid) and potential enstrophy (dashed) from an initial optimal σ_E structure at $k = 2.3$. Upper two curves are from the inviscid case; lower curves from the viscid case.

6 Conclusions

By the measure of growth rate in the energy or potential enstrophy norms, optimal structures have been found which are significantly more unstable than normal modes, because of their freedom to grow at different rates in different places. Normal modes, which grow at the same rate everywhere, appear in this perspective as severely-constrained optimal structures Indeed, optimal perturbations can grow (at least initially) even where there is no normal mode instability at all.

An intuitively appealing feature of the optimally unstable structures is that they vary only slightly in N-layer models as N is changed. Normal mode growth rates and structures, on the other hand, can be very sensitive to the value of N in N-layer models [4, 5] because of resolved levels falling near critical levels. In our work the concepts of phase speed and critical levels only arise in the specially constrained case of normal modes.

The mathematical approach employed in this study can be applied to a wide range of instability problems, including the instability of initial conditions in forecast models.

References

[1] Green, J.S.A, 1960, *Quart. J. Roy. Met. Soc.*, 86, 237-251.
[2] Farrell, B., 1989; *J. Atmos. Sci.*, 46, 1193-1206.
[3] Goldstein, H., 1950. *Classical Mechanics*. Addison-Wesley, 399pp.
[4] Staley, D.O., 1986; *J. Atmos. Sci.*, 43, 1817-1832.
[5] Bell, M.J., and A.A. White, 1988, *J. Atmos. Sci.*, 45, 1731-1738.

PREDICTING TIME SERIES USING A NEURAL NETWORK AS A METHOD OF DISTINGUISHING CHAOS FROM NOISE

J. B. Elsner
Department of Meteorology, Florida State University
Tallahassee, FL 32306, USA

Abstract A neural network approach is presented for making short-term predictions on time series. The neural network does better at short term predictions of a chaotic signal than does an optimum autoregressive model. Also the neural network is clearly capable of distinguishing between chaos and additive noise.

I. INTRODUCTION

One of the basic tenets of science is making predictions. If we know previous behavior, how can we predict the future behavior. The approach in many sciences requires two steps; construct a model based on theoretical considerations and use measured data as initial input. Since in many cases the underlying theoretical principles are known, model construction has been and continues to be a primary area of interesting research.

One class of alternative approaches is to build models directly from the available data. For these methods the data, given as a time series, is usually considered a single realization of a continuous random process. This is appropriate when the randomness is a result of complex interactions involving many independent and ultimately irreducib'• degrees of freedom. Along these lines, linear models have had some success especially in regards to relating cause and effect to physical phenomena, however, their predictive power is limited. The limitation is perhaps related to the inability to model the evolutionary dynamics of the system [1].

In the last decade advances in the theory of dynamical systems have demonstrated the existence of dissipative systems whose trajectories that depict their asymptotic final states are not confined on limit cycles (periodic evolutions) or tori (quasi-periodic evolution) but in submanifolds of the total available phase space which are not topological. These submanifolds are fractal sets and are often called strange attractors. The corresponding dynamical systems are called chaotic systems and their trajectories never repeat. Thus, their evolution is aperiodic but completely deterministic. Because the evolution is aperiodic any "signal" measured from a chaotic dynamical system "looks" quite irregular and exhibits frequency spectra with energy at all wavelengths (broadband spectra) similar to those of random "signals". Another important property of chaotic dynamical systems and their strange attractors is the divergence of initially nearby trajectories. Due to the action of the attractor the evolution of the system from two (or more) nearby initial conditions will soon become quite different. Since the measurement of any initial condition is subjected to some error such a property imposes limits on long-term prediction. Nevertheless, for a short time nearby trajectories may not diverge significantly and thus even though each individual evolution might be quite complex, the knowledge of the dynamics and especially of the structure of the attractor (dimensions, Lyapunov exponents, etc.) may prove beneficial to the art of short-term prediction.

Motivated by the above ideas, very recently a number of techniques for making predictions have been developed to exploit the underlying determinism in complex systems [2,3,4]. The purpose of this paper is to show that neural networks are capabable of making short-term predictions on time-series data that are better than an optimum autoregressive model and to show that such a methodology is capable of distinguishing chaos from noise.

II. EXAMPLES

In this section we present two examples showing the effectiveness of using a neural network for making predictions on time-series data. In general neural networks work by iteratively solving for a weight matrix which is used for the inner product that takes inputs to outputs. For time-series prediction, the inputs are taken as lagged values of the discrete time sequence. More details concerning neural networks are given in Rumelhart et al. [5] and Owens and Filkin [6].

The neural network architecture we employ consists of three layers, one input, one hidden, and one output layer (fig. 1).

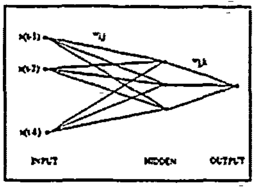


Figure 1. Architecture of the neural network used in the study.

Learning is achieved using back-propagation of errors resulting from the difference between predicted and the actual values during training [6]. Both of the time series used in this study consists of 1000 data points. Training is performed on the first 500 values with subsequent predictions made on the remaining 500 values. The number of input nodes is set at eight, the number of hidden nodes is set at three and the number of ouput nodes is set at one. Some trial runs indicated that the accuracy of prediction was not sensitive to small changes in the number of input or hidden nodes. The single ouput node represents some future value of the time series we wish to predict. Each training pattern consists of successive time-delayed values of the series similar to the method used by Perrett and van Stekelenborg [7] to predict annual sunspot numbers. For example, if we represent the series as x(t) where t=1, 2, . . . , 1000, then the first training pattern is {x(1), x(2), . . . , x(8)} and the ouput we are trying to predict is {x(9)}. Similarly, the second training pattern is {x(2), x(3), . . . , x(9)} and the ouput we are trying to predict is {x(10)}. Training continues over all input patterns for several thousand iterations.

For the first example a time series is taken from numerically integrating the Lorenz system [8] consisting of three ordinary differential equations describing convection of a fluid warmed from below in time. The time series of convective motion after all transients have diminished is shown in fig. 2a. Positive values indicate upward motion in the fluid. We take 1000 values from the time series, train the network on the first 500 values and make predictions on the last 500 values. Results of the neural network at predicting one step into the future (points) compared with the actual values (connected line) are given in fig. 2b. The normalized root-mean-square error (RMSE) between the actual and predicted values is 0.072 where zero implies a perfect forecast. Clearly the network is capable of capturing the underlying chaotic dynamics of the system.

To assess the predictive ability of the neural network against that of a standard statistical model we fit the first half of the time series using an optimum autoregressive process and then compare predictions on the second half of the series from both models. For the autoregressive model the time series is viewed as a single realization of a stochastic process which is taken to be stationary and having a Gaussian distribution. Employing the method of Katz [9] the optimum autoregressive model for the time series was found to be of 12th order.

Comparisons between the models are made by quantifying how the prediction accuracy (skill) decreases as predictions are

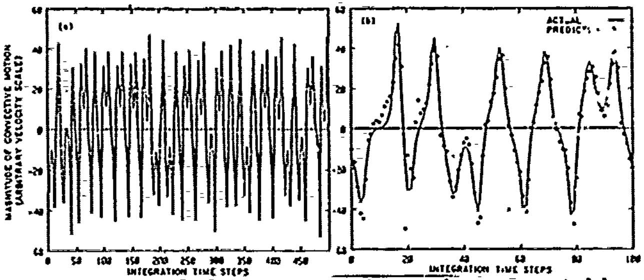


Figure 2 a. Time series of convective motions generated by numerically integrating the Lorenz system. b. Comparison of the actual time series (continuous line) with a neural network prediction (dots).

made further into the future. To do this we make a prediction one step into the future and then use this predicted value as one of the lagged inputs for the next prediction two time steps into the future. Similarly, the prediction at this second time step as well as the previous time step are used as lagged inputs for the next prediction three time steps into the future. Doing this successively allows us to compute the correlation coefficient between actual and predicted values as a function of prediction time where prediction time is given as discrete time steps into the future. The correlation coefficient between actual and predicted values is defined in the standard statistical way and is widely used as a measure of predictive skill. This procedure is followed for both the neural network model and for the optimum autoregressive model. Results are shown in Fig. 3a. For the first few steps into the future predictions from both models are good and the difference between the two models in terms of predictive skill is small. However, the neural network makes significantly better forecasts than does the autoregressive model as prediction time increases. The neural network model maintains greater predictive skill compared with the autoregressive model throughout the entire forecast period. The autoregressive model is essentially a linear model and therefore incapable of capturing the inherent nonlinear nature of such a record. Since the signal is, in fact, chaotic we cannot hope to make accurate predictions with any model too far into the future and we see the predictive skill of the neural network also drops to near zero after a relatively short time.

Recently it has been suggested that certain nonlinear prediction techniques are capable of distinguishing between chaos and noise in time-series records [4]. We demonstrate this capability with neural networks by comparing the above results with results from a model trained on a time series generated from discrete points on a sine wave having a unit amplitude and adding to it at each step a uniformly distributed random variable in the interval [-0.5, 0.5]. Such an example may display dynamical character similar to chaotic systems. For example, spectrum analysis will result in spectra exhibiting peaks superimposed an a continuous background and dimensional analysis may indicate anything from a low-dimensional system (if noise is weak) to a random signal (if noise is strong).

After training the neural network on the first half of the signal composed of a sine wave plus noise we make predictions on the second half and, as was done with the Lorenz system, we compute the correlation coefficient between actual and predicted values as a function of prediction time. The nearly horizontal line in fig. 3b is the result of this procedure. The independence of predictive skill with prediction length for this example is in sharp contrast to the rapid decrease of predictive skill for a chaotic signal (also shown in fig. 3b). From the differences we suggest that predicting time series using neural networks is another method for differentiating additive noise from deterministic chaos. With a simple autoregressive model, predictive skill on a time series containing periodicities and/or noise will show a marked dependency on prediction length making them inappropriate for distinquishing chaos.

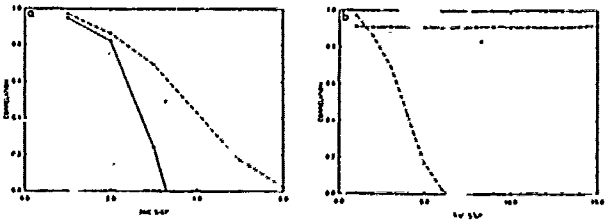


Figure 3 a. Correlation coefficients computed between actual values and predicted values as a function of prediction time for the convective motions using a neural network model (dashed line) and using a optimum autoregressive model (solid line). b. Correlation coefficients computed between actual values and neural-network predicted values as a function of prediction time for the convective motions (dashed line) and for the wave plus noise (solid line).

III. CONCLUSIONS

In applying chaos theory in the analysis of time-series data one usually begins by estimating the dimension of the underlying attractor [10, 11, 12, 13]. This is done by constructing a state-space embedding from the time series and then applying some variant of the correlation algorithm [14]. The dimension, which is given by the power-law (scaling) behavior of the correlation integral, gives a measure of the effective number of degrees of freedom of the system. Because the scaling regions used to estimate the dimension involve only a small number of distances between points in the state space much of the information in the time series is lost which for relatively short series can cause serious problems. In addition, such methods cannot in general distinguish output from a random process from output as a result of a chaotic dynamical process [15]. In this paper we show the ability of neural networks in making time-series predictions and demonstrate that such methodology is capable of distinguishing between additive noise and chaos.

Thanks are extended to Jay Perrett and John van Stekelenborg for introducing me to the back-propagation neural network and to A. A. Tsonis for stimulating discussions during the course of this research.

REFERENCES

[1] J. D. Farmer and J. J. Sidorowich, *Phys. Rev. Lett.* 59 (1987) 845.

[2] J. D. Farmer and J. J. Sidorowich, Theoretical Division, Center for Nonlinear Studies, Los Alamos Nat. Lab., Los Alamos, NM, 87545, LA-UR-88-901 (1988).

[3] M. Casdagli, *Physica D* 35 (1989) 335.

[4] G. Sugihara and R. M. May, *Nature* 344 (1990) 734.

[5] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Nature*, 323 (1986) 533.

[6] A. J. Owens and D. L. Filkin, Int. Conf. on Neural Networks, Washington D. C. 2 (1989) 381.

[7] J. C. Perrett and J. T. P. van Stekelenborg, Bartol Research Institute, Univ. of Delaware, Newark, DE (1990).

[8] E. N. Lorenz, *J. Atmos. Sci.* 20 (1963) 130.

[9] R. W. Katz, *J. Atmos. Sci.* 39 (1982) 1445.

[10] C. Nicolis and G. Nicolis, *Nature* 311 (1984) 529.

[11] K. Fraedrich, *J. Atmos. Sci.* 43 (1986) 419.

[12] C. Essex, T. Lookman and N. A. H. Nerenberg, *Nature* 326 (1987) 64.

[13] A. A. Tsonis and J. B. Elsner, *Nature* 333 (1988) 545.

[14] P. Grassberger and I. Procaccia, *Phys. Rev. Lett.* 50 (1983) 346.

[15] A. R. Osborne and A. Provenzale, *Physica D* 35 (1989) 357.

INITIALIZATION OF THE HIRLAM MODEL USING A DIGITAL FILTER

Peter Lynch,
Meteorological Service,
Glasnevin Hill,
Dublin 9,
Ireland.

and

Xiang-Yu Huang,
Department of Meteorology,
Stockholm University,
Arrheniuslaboratory,
S-106 91 STOCKHOLM,
Sweden.

Spurious high frequency oscillations occur in forecasts made with the primitive equations if the initial fields of mass and wind are not in an appropriate state of balance with each other. These oscillations are due to gravity-inertia waves of unrealistically large amplitude; the primary purpose of initialization is the removal or reduction of this high frequency noise by a delicate adjustment of the analysed data. In this paper a simple method of eliminating spurious oscillations is presented. The method uses a digital filter applied to time-series of the model variables generated by short-range forward and backward integrations from the initial time.

The digital filtering technique is applied to initialize data for the HIRLAM model. The method is shown to have the three characteristics essential to any satisfactory initialization scheme: (a) high frequency noise is effectively removed from the forecast; (b) changes made to the analysed fields are acceptably small; (c) the forecast is not degraded by application of the initialization.

The digital filtering initialization (DFI) technique is compared to the standard nonlinear normal mode initialization (NMI) used with the HIRLAM model. Both methods yield comparable results, though the filtering appears more effective in suppressing noise in the early forecast hours. The computation time required for initialization is about the same for DFI and NMI. The outstanding appeal of the digital filtering technique is its great simplicity in conception and application.

GRID SIZE INDEPENDENT CONVERGENCE OF THE MULTIGRID METHOD APPLIED TO FIRST ORDER PDES.

Per Lötstedt
Saab Aircraft Division, SAAB–SCANIA, S–581 88 Linköping, Sweden,
and
Dept. of Scientific Computing, Uppsala University, Sturegatan 4B, S–752 23 Uppsala, Sweden.

Abstract.
The multigrid method for numerical solution of first order partial differential equations is analyzed. The smoothing iterations are of polynomial type and are used on all grids, also the coarsest. It is shown that the convergence is independent of the grid under certain conditions. This is a result of a propagation of smooth error waves out of the computational domain and a damping of oscillatory error modes. Numerical experiments illustrate the theoretical results.

Introduction.
When the multigrid method is used for the numerical solution of systems of first order partial differential equations such as the Euler equations of fluid flow, it is not necessary to solve the equations exactly on the coarsest grid to obtain good convergence characteristics, see e g [3]. For elliptic equations all proofs of grid independent convergence assume that the solution is determined exactly on the coarsest grid [2]. Here we explain why the behaviour of the multigrid method is different for the two types of equations and prove that under certain sufficient conditions the convergence rate is independent of the grid also for first order equations.

Wave propagation and damping.
The multigrid strategy adopted here is a V cycle with a few pre and postsmoothing steps before going to and after returning from the next coarser grid. Also on the coarsest grid a few smoothing steps are taken. Let the discretization of our partial differential equation be

$$Qu = f.$$

For the convergence of linear problems it is sufficient to study the case f=0, which is equivalent to looking at the convergence to 0 of the initial error. The relaxation scheme S on grid l is assumed for smooth functions to be such that

$$\hat{S}_l = I - \Delta t_l \hat{Q}_l + O(\Delta t_l^2),$$

where a hat denotes the Fourier transform and Δt_l is a small parameter like a time step. Iterative methods of this kind are Runge Kutta time stepping [3] and the Chebyshev method [7]. For systems of partial differential equations with constant coefficients in two space dimensions it is shown in [5] by means of Fourier analysis that oscillatory parts of the error on the finest grid are damped efficiently but the smooth parts of the error are not damped very well. The difference between first order and elliptic pdes is that in the first case the smooth part of the error is propagated out through open boundaries This propagation is achieved with a time step

$$\Delta t_* = \sum_{l=0}^{L} \Delta t_l,$$

where L+1 is the number of grids. This was proved in [1] for L=1, one space dimension and a scalar equation. It is shown in [4,5] that the behaviour of the smooth part of the error is governed by the differential equation. Let us keep the coarsest grid fixed and add finer grids by halving the step size. Then the smooth part, which is well represented on all grids, is transported out through the computational boundaries at a rate which is determined by the differential equation and the accumulated time–step. A lower bound on this rate is given by the rate on the coarsest grid. Furthermore, the damping of the oscillatory error modes can be made independent of the finest grid. In [4] an upper bound independent of the finest grid size is derived for the number of V–cycles needed for a two–dimensional scalar, constant coefficient first order pde to converge with the multigrid method. The results can be generalized to one and three space dimensions and to the W–cycle.

Numerical experiments.
A central difference approximation of the space derivative with added artificial viscosity and a Runge–Kutta time–stepping procedure as in [3] are used in the first two examples. The coarse grid with 24 points is the same in the grid sequences in the examples and the step size is doubled between a grid and the next coarser grid. There is one presmoothing step in the V–cycle. In fig. 1 a simple one–dimensional example is displayed. The equation is

$$u_x = 0,$$

and the initial error moves to the left through the boundary as the iterations proceed. The smooth error pulse is plotted every 5:th iteration. It is damped and propagated at approximately the same rate with 1, 2, 3 and 4 grids. In [4] similar results are obtained with the Chebyshev method. The Euler equations of fluid flow are solved in a two–dimensional channel with a bump using the grids 40x12, 80x24 and 160x48 in fig. 2. The upper and lower walls are solid and the left and right boundaries are open. The Mach number of the flow from the left is 0.5. The steady state solution on the intermediate grid is shown and the rate of convergence is measured by the residual of the continuity equation (cf. [3]). No attempt has been made to optimize the performance of the method. Finally, in fig. 3 three steps of the GMRES method [6] are used as iterative smoother in the multigrid method. The equation and the discretization are the same as in fig. 1 and the solution is plotted in every step. The results with the initial bell shaped error pulse are similar to those in fig. 1 with a convergence almost independent of the grid size even if GMRES does not satisfy the conditions on the relaxation scheme with a constant time–step.

References.

1. B. Gustafsson, P. Lötstedt, Analysis of the multigrid method applied to first order systems, Proc. 4th Copper Mountain Conf. on Multigrid Methods, Eds. J. Mandel et al, SIAM, Philadelphia, 1989, 181–233.

2. W. Hackbusch, Multi–Grid Methods and Applications, Springer Verlag, Berlin, Heidelberg, 1985.

3. A. Jameson, Computational transonics, Comm. Pure Appl. Math.. XLI (1988), 507–549.

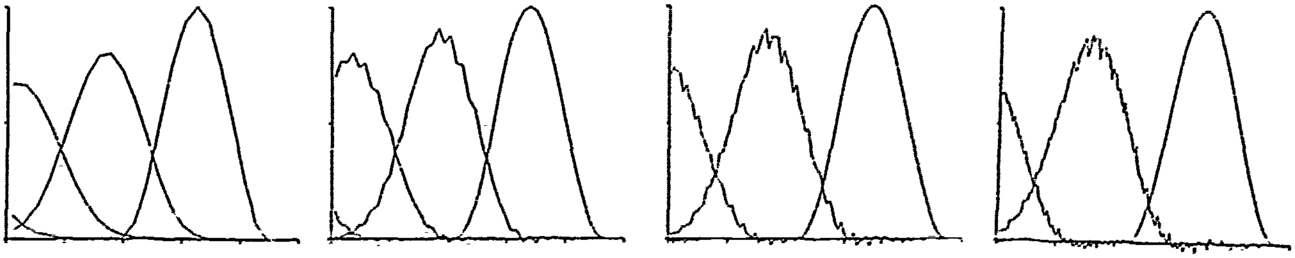
4. P. Lötstedt, Grid independent convergence of the multigrid method for first order equations, Report to appear, Dept. of Scientific Computing, Uppsala University, Uppsala, 1991.

5. P. Lötstedt, B. Gustafsson, Fourier analysis of multigrid methods for general systems of PDE, Report 129, Dept. of Scientific Computing, Uppsala University, Uppsala, 1990.

6. Y. Saad, M. H. Schultz, GMRES. A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 7 (1986), 856–869.

7. R. S. Varga, Matrix Iterative Analysis, Prentice–Hall, Englewood Cliffs, NJ, 1962.

Figures.



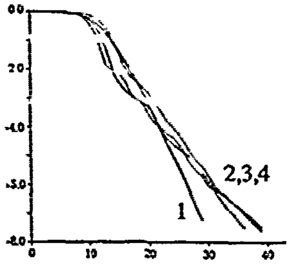
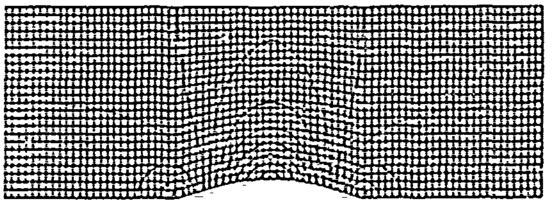


Fig. 1. The initial bell shaped error moves to the left as the multigrid iterations with Runge–Kutta time–stepping proceed (above). Eventually it disappears through the left boundary. The number of grids is from left to right. 1, 2, 3 and 4. The coarse grid is the same. The logarithm of the Euclidean norm of u as a function of the number of V–cycles is presented showing the grid independence (left).



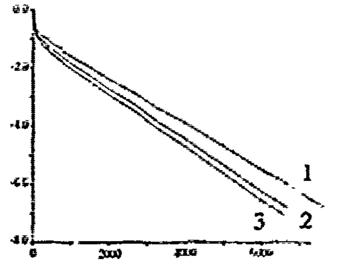
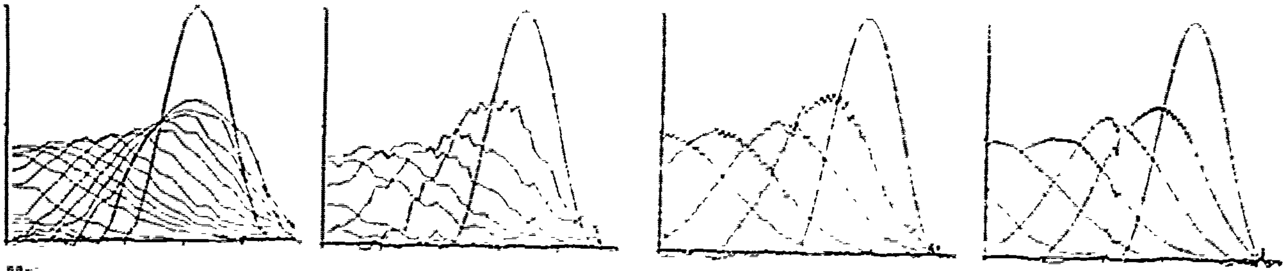


Fig. 2. The intermediate grid and isobars of the Euler solution are displayed to the left. The convergence history for the multigrid method with Runge–Kutta time–stepping for 1, 2 and 3 grids is shown to the right. The coarse grid is the same in all cases. The logarithm of the Euclidean norm of the residual is plotted vs the number of V–cycles.



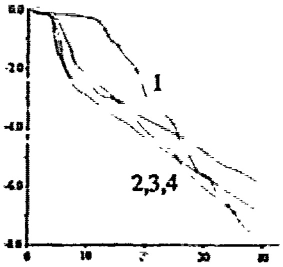


Fig 3. The initial bell shaped error moves to the left as the multigrid iterations with GMRES proceed (above). Eventually it disappears through the left boundary. The number of grids is from left to right. 1, 2, 3 and 4. The coarse grid is the same. The logarithm of the Euclidean norm of u as a function of the number of V–cycles is presented showing the grid independence (left).

Numerical Solution of the Equations for Two Dimensional Compressible Reacting Fluid Flow

Björn Sjögreen
Uppsala University
Department of Scientific Computing
Sturegatan 4B
S-752 23 Uppsala
Sweden

Abstract: We solve numerically a system of partial differential equations describing the motion of a gas in which a chemical reaction occurs. We investigate detonation wave solutions in two space dimensions. Poor resolution of the reaction layer will lead to problems due to stiffness of lower order terms. We demonstrate such a failure and suggest a remedy for it.

1. Problem to Solve

We here consider the equations

$$\begin{pmatrix} \rho \\ m \\ n \\ e \\ \rho z \end{pmatrix}_t + \begin{pmatrix} m \\ m^2/\rho + p \\ mn/\rho \\ m(e+p)/\rho \\ mz \end{pmatrix}_x + \begin{pmatrix} n \\ mn/\rho \\ n^2/\rho + p \\ n(e+p)/\rho \\ nz \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -K\rho z e^{-T_i/T} \end{pmatrix} \qquad (1)$$

describing a fluid in which a one step irreversible chemical reaction is taking place. $\rho(x,y,t), m(x,y,t), n(x,y,t), e(x,y,t)$ are the density, x- and y-momentum and energy of the gas. $z(x,y,t)$ is fraction unreacted gas. The pressure is given by

$$p = (\gamma - 1)(e - \frac{1}{2}(m^2 + n^2)/\rho - q_0 \rho z)$$

and the temperature is defined as $T = p/\rho$.

We restrict ourselves to two space dimensions and we have neglected heat transfer and viscos effects. We will focus on discountinuous solutions, detonation waves. For these waves the viscosity is not as important as for the slower deflagration wave solutions. The solutions computed have qualitative features agreeing with experimental data.

q_0, T_i, γ, K are parameters, named chemical heat release, ignition temperature, c_p to c_v ratio, and equilibrium constant respectively.

2. Numerical Method

The solution to (1) is in general discontinuous. Numerical methods specifically designed for such solutions have been developed in recent years [1,2]. We use a TVD method of second order accuracy to approximate the convective fluxes [3,4], the right hand side is solved implicitly by the trapezoidal rule.

In one space dimension, we introduce the grid points $x_j, j = \ldots, -2, -1, 1, 2, \ldots,$ and the time levels $t_0, t_1, t_2, \ldots$. The grid spacing is $\Delta x = x_j - x_{j-1}$ and the time step $\Delta t = t_n - t_{n-1}$. For the equation

$$u_t + f(u)_x = g(u) \qquad -\infty < x < \infty \quad t > 0 \qquad (2)$$

the method then takes the form

$$u_j^{n+1} = u_j^n - \lambda \Delta_+ h_{j-1/2}^n + \Delta t \frac{1}{2}(g(u_j^{n+1}) + g(u_j^n))$$

here, $\lambda = \Delta t/\Delta x$ and $h_{j+1/2}^n = h(u_{j+2}^n, u_{j+1}^n, u_j^n, u_{j-1}^n)$ is the numerical flux function, satisfying the consistency condition

$$h(u,u,u,u) = f(u)$$

u_j^n is the approximation to the solution at the point (x_j, t_n). The spatial difference operator is defined as $\Delta_+ a_j = a_{j+1} - a_j$ We solve the implicit equation for u_j^{n+1} using Newton's method.

The approximation in two space dimensions is similar, but for the usage of a two dimensional grid $(x_j, y_k), j = \ldots, -1, -2, 0, 1, \ldots, k = \ldots, -1, -2, 0, 1, \ldots$.

3. Some Results and Failures

The following values of the parameters were used in the computations. $T_i = 50, K = 10000, q_0 = 50, \gamma = 1.2$. The computational domain was $1 \times \frac{1}{2}$ in size and a grid was moved along with the detonation wave.

Initially a Chapman-Joguet wave was started along a slightly curved front. Figure 1 shows a solution computed on a 100×50 grid, using a cfl number of 0.4 The solution is displayed at six different times, to show how the triple points move back and forth. This behavior leads to the cellular pattern observed in experiments [5].

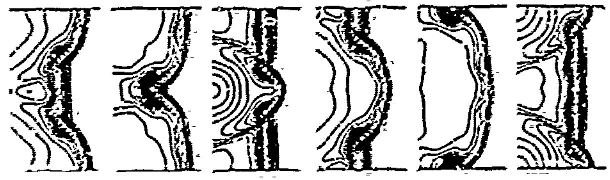


figure 1, density contour lines

Figure 2 shows the same solution computed with a twice larger time step, i.e. cfl=0.8. For this time step the method is stable according to linear stability analysis. At a certain time the triple points cease to move and a pattern with a triangular shape extending from wave front emerges. The solution then stays that way for all times. No movement of the triple points in the y direction is observed This is not a realistic solution.

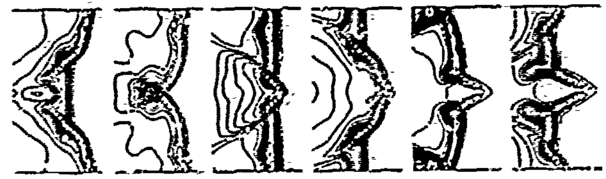


figure 2, density contour lines

It is well known that difficulties due to stiffness is encountered for $K\Delta t$ large, even when using an implicit method for the right hand side. See [6,7] for discussions of the one dimensional problem.

We can understand this difficulty by considering the last equation in (1). The approximation described above will lead to the expression

$$(\rho z)_j^{n+1} = \frac{(\rho z)_j^n - \lambda \Delta_+ h_{j-1/2}^n}{1 + \Delta t K e^{-T_i/T}} \qquad (3)$$

here we simplify, by neglecting the dependency of T on ρz, (this simplification is not done in the computations.) If the temperature increases near T_i, and if $K\Delta t$ is large, $(\rho z)_j^{n+1}$ will immediately drop to zero. A small change in temperature triggers a large change in z which in its turn effects the temperature through the third equation in (1), the result can become a wave traveling at the unphysical speed one grid point per time step.

4. An Improved Method

A detonation wave consists of a non reacting shock wave which first increases the temperature of the fuel mixture, so that ignition occurs behind the shock wave. No chemical reactions start before the shock wave has passed through. We try to emulate this behavior in the numerical method Numerically, there are always a few grid points in the shock. We want to make sure that none of the points inside the shock triggers the chemistry. The simplest way of ensuring this is to evaluate the right hand side a few grid points ahead of the shock. This method has been successful in one space dimension. The main point of this paper is that the method also remedies the error in our two dimensional computation above.

Figure 3 shows the same computation as figure 2, with the exception that the right hand side is evaluated one grid point out from the wave front. The direction in which to go is determined as the direction of smallest temperature. i.e. in one space dimension we evaluate

$$(\rho z)_j^{n+1} = (\rho z)_j^n - \lambda \Delta_+ h_{j-1/2}^n - \Delta t K (\rho z)_{j+d}^{n+1} e^{-T_i/T_{j+d}^{n+1}}$$

with d the value of k that minimizes T_{j+k}^n over $k = -1, 0, 1$. The two dimensional case is the straightforward generalization of this procedure.

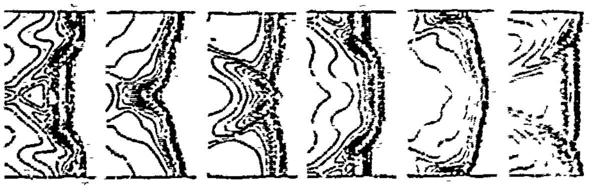


figure 3, density contour lines.

We plot density contours of the solution on a domain determined by the time- and y-axis. The same cellular pattern as seen on sooted plate recordings in experiments [5] appears. Figure 4, 5 and 6 show these plots for the cases discussed above. Figure 4 is computed with the small time step, figure 5 with the twice larger time step and finally figure 6 is obtained using the larger time step and the modification.

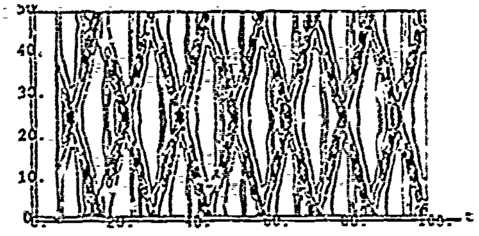


figure 4, the y-t plane

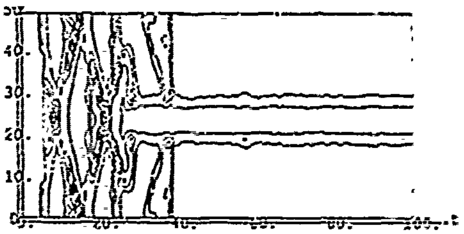


figure 5, the y-t plane

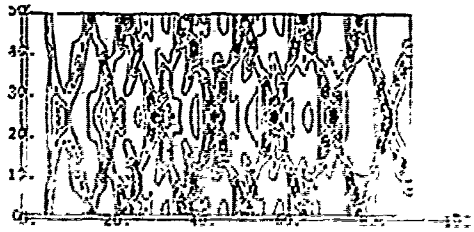


figure 6, the y-t plane

Both the run with small time step, and the modified method with larger time step predicts approximately the same size for the detonation cells.

References

[1] A Harten, "High Resolution Schemes for Hyperbolic Conservation Laws" J. Comput. Phys.. 49 (1983), pp. 357–393.

[2] A. Harten, S. Osher, B.Engquist, S.Chakravarthy "Some Results on Uniformly High-Order Accurate Essentially Nonoscillatory Schemes", Applied Numerical Mathematics 2 (1986), pp.347–377

[3] P L Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes", J. Comput. Phys., 43 (1981), pp. 357–372.

[4] P.Sweby. "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws", SIAM, J. Numer. Anal. 21 (1984), pp.995–1010.

[5] D.C Bull, J.E.Elsworth, P.J.Shuff, E.Metcalfe "Detonation Cell Structures in Fuel/Air Mixtures" Combust. Flame 45, (1982), pp.7–22.

[6] M.Ben-Artzi, "The Generalized Riemann Problem for Reactive Flows", J.Comp.Phys., 81, (1989), pp.70–101.

[7] P.Colella, A.Majda, V.Roytburd "Theoretical and Numerical Structures for Reacting Shock-Waves", SIAM J.Sci.Stat.Comput. 7, (1986), pp.1059–1080

A BLACKBOARD SOFTWARE DESIGN FOR CFD SOFTWARE

M. Petridis,B. Knight, D. Edwards
Centre for Numerical Modelling and Process Analysis
Thames Polytechnic
Wellington St
Woolwich, London SE18 6PF, UK

Abstract- This paper presents a software design approach for CFD software. The approach is based on the Data Structure design methodology and results to an architecture built around the Blackboard structure. An experimental prototype implementation of this approach is described particularly with reference to the Blackboard structure as well as the integration of an Intelligent Knowledge Based System in the form of a rulebase and a set of qualitative data entries.

I. INTRODUCTION

Computational Fluid Dynamics (CFD) software is a continuously developing area in engineering software which is particularly known for its highly algorithmic nature as well as the diversity of its application fields within science and engineering. Different areas such as aerodynamics, meteorology, thermodynamics and chemical reactions modelling are among those providing a long queue of new mathematical models for solution by a CFD software package.

The need for integration of all phases and aspects of the mathematical modelling process, from problem set-up to the final results validation and interpretation as well as the quest for more reliable and accurate solutions have increased the demands on CFD software in terms of complexity and efficiency. The increasing availability and accessibility of more powerful computer hardware in terms of memory and processing speed has not managed to quench the thirst for increased power of the CFD modelling community but on the contrary it has increased the appetite for solutions to even more complex modelling problems. An other aspect is that the number and the unpredictability of the changes in specifications and requirements that are usually expected during the long life-cycle of a CFD software product make the maintainability of the software imperative.

Finally, the interactive nature of modern CFD modelling software and the diversity of the end users of the software product ranging from the mathematician, the numerical analyst and the software analyst to the specialised scientist and engineer, all with different degrees of expertise in the use of the software, make the human interface a decisive factor in the overall performance of the software.

All these call for a fresh approach to the design of CFD software, using an overall strategy which has to be formally defined, with specific methods, tools and goals which can be referred to and used for validation of the final software product.

A Software design approach for CFD software

For the construction of an overall design approach for CFD software, it is necessary to identify the most important stages of the software development life-cycle where vital decisions will be taken influencing and determining the final quality of the software product.

We could highlight the following:

(a) The choice of a formal software design strategy- The common strategy used for CFD software design is the Data Flow design approach (Yourdon[1]) which views the overall software structure as based on the flow of data between a series of software modules that transform the data in a continuous fashion ultimately from input to output data. This view which has a taste of the old batch-job days, recognises the dominance of algorithms and processes over the actual structure of the data. Nevertheless, during the life-cycle of a CFD package, the algorithms, the various solution techniques and processes are much more likely to change than the structure of the database which represents the problem itself.

The highly physical and especially geometrical nature of the underlying mathematical model creates the need for an accurate and flexible representation of the real physical problem, that is, the specific portion of reality relevant to the problem. This leads to a Data Structure design approach (Jackson[2]), which is based on the conceptual database schema. This is the static, time-invariant and solution independent view of the problem, which can be the sound design foundation for the resulting software structure. The conceptual schema can be constructed using the Entity-Relationship approach(Chen[3],Knight[4]) and it can be shown that a useful partition can be made into problem and solution spaces(Petridis et al[5]). Using this schema we can design a flexible, well defined database By reference to this, a series of cohesive and uncoupled software modules can be built, leading to an overall software design based on the information hiding principle.

(b) Integration into a software architecture- Once the database and the software modules have been properly designed and defined, there is a need for an overall architecture that can provide the operational framework of the software package. This architecture must be defined in terms of

1. The control structure
2. The modular communications structure and mechanism
3. The human interface

(c) The design of the human(user) interface- The human interface must be able to respond to the user in a natural and friendly way. It should

- Communicate to the user using a set of terms and concepts in a form that is naturally perceived by him.

- Give a flexible and layered "feel" to the different users(in terms of background and expertise) that may use the software.

Integrate expert knowledge and inferencing procedures into the software.

- Give the ultimate control and decision making to the user.

One way of tackling these points is by integrating an Intelligent Knowledge Based System (IKBS) into the software. Typically IKBS systems are capable of delivering many of the requirements of such a design. They allow the integration of much of the expertise needed in such a design, in a particularly flexible and incrementable way Expertise exists in many different areas. For example in the solution method itself, in terms of speed and accuracy, and in terms of the user requirements of the software. what sort of solution does the user want? These requirements constitute what is afte. referred to as Intelligent Front Ends(Rissland[6]) to software. However it is argued here that this reference gives the wrong view of the architecture for intelligence, since it implies that it can be bolted on to existing software packages. The view we take here is that the architecture for the system must allow for a more integral IKBS component. Expertise exists at every level of CFD, from solution method to physical processing requirements, to user interface.

The integration of the software into the overall software architecture which was discussed in point (b) above, is the design decision that will affect most the resulting software quality mainly in terms of maintainability and efficiency. One such architecture which has shown considerable advantages for building CFD software packages is presented in the following sections.

The Blackboard architecture

The Blackboard architecture is a software framework that has its roots in Artificial intelligence (Morgan[7]). It is built around the Blackboard (B-B) construct which is used as the main mechanism for passing information and control between software modules. As its name indicates, the blackboard is basically a mechanism that stores several pieces of information and on which a number of software modules have restricted access to write and read. As shown in figure 1, modules A and B have both access to parts of the B B and so can selectively share information they need for their function. This is a flexible way of passing data from one module to an other, hiding if necessary information about the nature of these data, as each module knows where to find the information it needs. The Database Management System(DBMS) can be used as a filter of such information providing different modules with the data they request and in the form they need them.

The B-B architecture for CFD software

The Blackboard architecture was implemented in an experimental prototype CFD application which was built to demonstrate this approach to CFD software design. This is a working CFD package developed to deal with a characteristic family of CFD modelling applications. It is essentially a two dimensional transient scalar multiphase finite difference code containing among others a number of numerical solvers and grid generators that can be used to tackle a class of application problems. The code is implemented in the 'C' programming language and runs under various operating systems. The database and the software modules are designed using a Data Structure oriented approach and the whole software structure is built around the Blackboard architecture as shown in Figure 2.

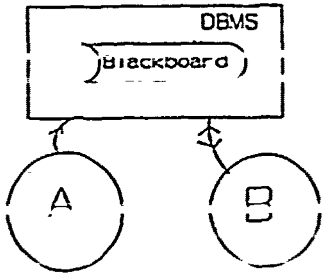


Figure 1 The Blackboard

The information stored on the Blackboard is an abstraction of the information stored in the working database, as the Blackboard provides an abstract overview of the problem and the solution for the given modelling application together with the state of the modelling process at any given point in time as well as the dynamic of the processes and tasks that are currently executing or are waiting to be executed. As shown in Figure 2, the Blackboard system comprises of the blackboard database around which a database management system is built to allow a flexible and polymorphic view of the Blackboard data from the accessing software modules. The Blackboard itself is implemented as a hierarchical structure containing information about the problem, the solution settings and state at any point and a tasks buffer containing a number of tasks that have been decided to "fire" but did not have the opportunity to do so yet. A task is implemented as a piece of data encapsulating a function and a number of data parameters or other functions that are needed for the realisation of the particular task. The operation of the Blackboard system is controlled by the main module M which looks at the solution state and the tasks buffer and if required sends a message to the appropriate module to come to the B-B system and have the particular task assigned to it. A series of modules (0-3) are working on the database and have restricted access to the Blackboard system. When a message is send to one of them by the B-B system, that module goes to the tasks buffer and takes the next task in it together with the information needed for its execution, leaving a message to notify that it has taken up the task. After completion of the assignment the module returns to the B-B system and reports the successful (or unsuccessful) termination of its task.

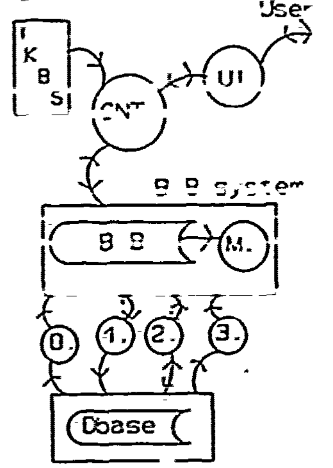


Figure 2 The Blackboard architecture for the CFD prototype

There are four different types of working modules.

- 0. Database Browsers - Pattern Recognisers that look through the amount of data in the database for patterns in the data to store in an abstract form in the B-B.
- 1. Database Constructors-Modifiers (such as grid generators) that construct the topology and instantiate the database in a dynamic fashion.
- 2. Database Updates (such as solvers) having restricted access to values of data in the database.
- 3. Output functions that collect values from the database and work on them in order to output to the user(such as contouring modules).

The control module CNT asks the user for his requirements and consults the Intelligent Knowledge Based System (IKBS) as well as looks at the B-B system to have an overview of the solution state at any time. Combining the above, the control module uses its built in inference engine to decide what the next task(s) to be performed is(are),outputs to the user the reasoning of this decision and puts the task(s) in the tasks buffer (B-B)

Discussion of the resulting architecture-Conclusions

This architecture has shown to possess a number of advantages:
It provides a flexible way of intermodular information and control passing.
It gives a natural separation and data uncoupling of the control and user interface modules from the number crunching modules which enhances flexibility and allows for an easier maintenance of the software because of the hiding of important decisions mainly with respect to the data structure.
-It contains an abstract representation of knowledge about the problem and its solution at any point as a substitute of the huge amount of raw data usually associated with CFD software. This information is in a form closer to the natural perception of the user and the way that expert rules are formulated. This makes the task of integrating an IKBS into the software easier.

An IKBS has been integrated in the experimental CFD software prototype comprising a Data Dictionary and a Rulebase which are accessed by the control module (CNT) (Figure 2) and interpreted by the Inference Engine which is built in the module.

The Data Dictionary is a collection of entries describing qualitative data formally defining and quantifying them. Concepts such as "Fine grid" or "slow convergence" which can be useful for the natural description of a state and the formulation of expert rules are defined in the Data Dictionary. The Rulebase is a collection of expert rules that are formulated as Prolog clauses and describe in a heuristic or production manner the knowledge that can be used to infer actions to be taken by the software during the various stages of the modelling process. So there are rules to decide the descretisation technique or the solver to be used and to monitor and accelerate the successful modelling process changing settings and parameters. Two examples follow:

- Setting up the solver

Is(Model,"Laplace") if Is(Time_dependency,"steady_state") and
Is(Phases,1) and
Solver("rectangular","line_SOR").

- Determining the successful conclusion of the solution procedure

End_solver if Is(Convergence,"acceptable").

The truth of predicates in the rules is determined in a usually recursive fashion looking up entries from the Data Dictionary and comparing them to qualitative data filtered from the working database. A set of tools has been integrated to the prototype comprising of Data Dictionary and Rules editor and a compiler which translates the data entries and the rules respectively to 'C' structures and 'C' recursive structures so that different Rule sets and Data entries sets can be embedded in the software to deal with different areas of CFD modelling.

There is obviously a higher communications overhead associated with the use of the B B system but due to the CPU intensive number crunching nature of CFD software this does not affect substantially the overall efficiency of the software but experience has shown that there are a lot to be gained by this approach to CFD software design although more work is needed in validating and extending the existing prototype and looking for ways to optimize its overall performance.

References

1. Yourdon, E., Constantine, L., Structured design, Prentice Hall,1979.
2. Jackson, M.A., Principles of Program Design, Academic Press, 1975
3. Chen P., The entity-relationship model towards a unified view of Data, ACM Transactions on database systems, Vol.1,No1,pp.9 36,1976.
4. Knight, B., A Mathematical Basis for Entity Analysis,in Entity Relationship Approach to Software engineering (ed. Davis,),pp 81 90,Proceedings of the Third International Conference on Entity-Relationship Approach, Anaheim, California, U.S.A., 1983, North Holland, 1983.
5. Petridis, M., Knight, B.,Edwards, D., A Design for reliable CFD Software, in Reliability and Robustness of Engineering Software II,(Ed Brebbia, C.A., Ferrante, A.J.),pp.3-17, Proceedings of the 2nd International Conference held in Milan,Italy,22 24 April 1991,Elsevier, 1991.
6. Rissland, E.L., Ingredients of intelligent user interfaces, Int. Journal Man-Machine Studies (1984) 21,377-388.
7. Morgan, T., Engelmore, R., Blackboard Systems, Addison-Wesley, 1988

The Role of Physical Analysis in the Numerical Simulation of Aerodynamic Flows

Hanxin Zhang
China Aerodynamic R & D Center
P. O. Box 211
Mianyang, Sichuan 621000
PRC

Fenggan Zhuang
Chinese Aerodynamic Research Society
P. O. Box 2425
Beijing, 100080
PRC

Abstract

Typical examples are given for the illustration of the role of physical analysis in the numerical simulation of aerodynamic flows. The first is related to the proper choice of grid Reynolds number and avoidance of spurious oscillations or even chaos in a numerical calculation of shock waves with shock capturing technique. The second shows the development of a vortex along its axis, the existence of unstable limit cycle and its relation to flow picture in the bursting region.

I. Introduction

With the rapid development of supercomputers it seems that numerical results are easily obtained The essential problem is that whether the results thus obtained are physically in reality and the large quantity of numerical data if improperly treated would lead to distorted physical flow pictures which may often be misleading Thus the treatment given in this paper may not be without practical significance. Examples are given for the study of flows containing shock-waves and vortex flows.

II. Solution of one dimensional shock wave

For steady flow in one dimension, the Navier Stoke's equations can be integrated yielding

$$\frac{du}{dx} = \beta\frac{(u-u_\infty)(u-u_2)}{u} \tag{1}$$

where

$$\beta = \frac{3(\gamma+1)}{8\gamma}\cdot\frac{p_\infty u_\infty}{\mu} \tag{2}$$

p_∞ and u_∞ are density and velocity at infinite upstream respectively, γ is the ratio of specific heats and μ the coefficient of viscosity, u_2 is the velocity at infinite downstream, which is just the inviscid velocity behind a normal shock wave Of course here we assume u_∞ is a supersonic velocity. If we take u_∞ as the reference velocity, equation (1) becomes

$$\frac{du}{dx} = \beta\frac{(u-1)(u-u_2)}{u} \tag{3}$$

where

$$u_2 = \frac{\gamma-1}{\gamma+1}(1+\frac{2}{(\gamma-1)M_\infty^2}) \tag{4}$$

In order to reveal the possibility of producing chaos we first use perturbation technique and keep only second order terms, we can see immediately the equation is reduced to Landau's equation Let us suppose the initial value be given at far upstream and we are looking forward the solution as we march downstream. Then it is possible to put

$$u = 1+\varepsilon \tag{5}$$

Substituting eq.(5) into eq.(3) and keeping terms up to $O(\varepsilon^2)$ we have

$$\frac{d\varepsilon}{dx} = \beta(1-u_2)\varepsilon+\beta u_2\varepsilon^2 \tag{6}$$

which is indeed a Landau equation. Write down corresponding finite difference equation in the following form

$$\varepsilon^{n+1} = \varepsilon^n + \beta\Delta x(1-u_2)\varepsilon^n + \beta\Delta x u_2(\varepsilon^2)^n \tag{7}$$

where $\Delta x = x^{n+1} - x^n$ Introducing a new variable z

$$z = -\frac{u_2\beta\Delta x}{1+\beta(1-u_2)\Delta x}\varepsilon \tag{8}$$

We have

$$z^{n+1} = \alpha_u z^n(1-z^n) \tag{9}$$

and

$$\alpha_u = 1+\beta(1-u_2)\Delta x \tag{10}$$

We immediately recognize that if $\alpha_u > 3$, i.e.

$$\beta\Delta x > \frac{2}{1-u_2} = r_u^* \tag{11}$$

period doubling will occur. We note $\beta\Delta x$ is essentially the grid Reynolds number, r_u^* depends on γ and M_∞, take for exampie $\gamma = 1.4$ and $M_\infty = 4$, then $r_u^* = 2.56$. The period doubling pheonomenon has been verified by numerical simulation of one-dimensional Navier Stokes equation. Similar conclusion can be drawn if we start the solution at downstream side and marching toward the upstream.

III. The Development of a Vortex along its Axis

Suppose we have a vortex spiralling around its axis z We now study the flow pictures in the transversal planes, where x, y and z form a local cartesian coordinates with the origin at o. Along z axis the transversal velocity components u and v are zero. Let us consider that the flow is steady, the equation for the streamline is

$$\frac{dx}{u(x,y,z)} = \frac{dy}{v(x,y,z)} = \frac{dz}{w(x,y,z)} \tag{12}$$

The following results can be easily obtained.

1. If $\partial(\rho w)/\partial z$ has the same sign in the neighborhood of the origin (either always positive or always negative), then in this region the transversal section streamlines will not be closed.
2. Depending on whether $\partial(\rho w)/\partial z$ is positive or negative, we will have the transversal streamlines being spiral outward or inward, and in the usual terminology we call spiral inward vortex as stable and spiral outward vortex as unstable.
3. At the point of a vortex bursting $\rho w = 0$, so there must be a decellerating flow just before bursting to occur and vortex bursting is accompanied with an unstable spiral.
4. From a similar analysis in nonlinear mechanics, we know there exists a stable limiting cycle in front and in the vortex bursting region. In case we have a bursting bubble, then there must exist a second limiting cycle, though this limiting cycle is unstable. So there is no guarantee for the actual realization of steady vortex bursting bubble.
5. In the longitudinal section at bursting point, the section streamlines are of saddle type.

We have made a systematic analysis of available experimental data related to vortex bursting and also made a study of flow around an axisymmetric obstacle placed on a flat plate. The numerial results agree with all qualitative pictures mentioned above

IV Conclusions

Though the physical analysis made so far involving little mathematics yet it already shows the potential involved to understand complex flow pictures, otherwise we will be lost in an ocean of numerical data. It is also essential to uncover the underlying physical mechanism and to lead to effective means to control or modify the flow to suit our specific purposes.

The research was supported by the China National Natural Science Foundation.

UPWIND COMPACT SCHEME WITH DISPERSION REGULATOR[1]

Ma Yanwen   Fu Dexun
Laboratory for Nonlinear Mechanics of Continuous Media
Institute of Mechanics, Chinese Academy of Sciences, Peijing 100080

Abstract-For improving the resolution of the shock. a dispersion regulator and an artificial dispersion are introduced, and an upwind compact scheme with dispersion regulator is developed.

1. INTRODUCTION

In ref.1 a compact scheme was developed.This scheme was useful for improving both of the accuracy and the efficiency, but there are oscillations in the numerical solutions. In this paper a dispersion regulator and artificial dispersion are introduced and an upwind compact scheme with dispersion regulator is developed. Numerical experiments show the scheme developed has high resolution.

2. UPWIND COMPACT SCHEME WITH DISPERSION REGULATOR

A. Compact Scheme and Upwind Compact Scheme

Consider the model equation

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \qquad f = au \quad , \quad a = \text{const.} \tag{1}$$

In ref.1 a compact scheme is developed. The scheme has high accuracy, but there are oscillations in the numerical solutions near the shock. The solutions may be improved if the scheme is upwind biased. Consider a semi-discrete approximation for equation (1)

$$\frac{\partial u}{\partial t} + \frac{1}{\Delta x} F_j = 0 \tag{2}$$

$$\alpha_0 F_{j+1} + \beta_0 F_j + \gamma_0 F_{j-1} = a_0 \delta_x^+ f_j + b_0 \delta_x^- f_j \tag{3}$$

$$\alpha_0 = \frac{1}{6} + \varepsilon_2 - \varepsilon_3 - \varepsilon_4 \quad , \qquad a_0 = \frac{1}{2} - 2\varepsilon_4$$

$$\gamma_0 = \frac{1}{6} - \varepsilon_2 - \varepsilon_3 + \varepsilon_4 \quad , \qquad b_0 = \frac{1}{2} + 2\varepsilon_4$$

$$\beta_0 = \frac{2}{3} + 2\varepsilon_3$$

where δ^+, δ^- and δ° are forward, backward and central differences. Scheme (2) and (3) for $a>0$ with $\varepsilon_2 > 0$ and $\varepsilon_4 > 0$ is called upwind compact scheme. With different choice of the parameters ε_k different schemes can be obtained. The simplest one within schemes with accuracy of order three is

$$\frac{2}{3} F_j + \frac{1}{3} F_{j-1} = \frac{1}{6} \delta_x^+ f_j + \frac{5}{6} \delta_x^- f_j \tag{4}$$

It is dissipative and it can be solved easily.

B. Upwind Compact Scheme with Dispersion Regulator

Consider a semi-discrete second order accurate scheme

$$\frac{\partial u_j}{\partial t} + \frac{1}{\Delta x} \delta_x^\circ f_j = 0 \tag{5}$$

The modified equation with the leading term takes form

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \Delta x^2 \frac{\partial}{\partial x}\left(R \frac{\partial^2 u}{\partial x^2}\right) \quad , \ f = au \ , \ R = \text{const} \tag{6}$$

Rewrite it into form

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \Delta x^2 \frac{\partial}{\partial x} \mu \frac{\partial u}{\partial x} \ , \quad \mu = R \frac{\partial^2 u}{\partial x^2} \Big/ \frac{\partial u}{\partial x}$$

If we have a shock as shown in Fig.1 the coefficient $\mu > 0$ in the front of the shock. Negetive μ is a reason of oscillation production behind the shock in the numerical solutions. In the present paper the parameter R in (6) is reconstructed as

$$R(u) = |\sigma a| \frac{|u(x+\Delta x) - u(x)| - |u(x) - u(x-\Delta x)|}{|u(x+\Delta x) - u(x)| - |u(x) - u(x-\Delta x)|} \tag{7}$$

which is called dispersion regulator in this paper. The function R(u) changes sign across the shock.

With dispersion regulator we have two kinds of schemes:

a. Scheme with 'artificial dispersion. The scheme suggested is

$$\frac{\partial u_j}{\partial t} + \frac{1}{\Delta x} \delta_x^\circ f_j = \frac{1}{\Delta x} \left\{ \delta_x^\circ (R \delta_x^2 u_j) - \frac{1}{2} \delta_x^2 (|R| \delta_x^2 u_j) \right.$$

b. Upwind compact scheme with dispersion regulator. The parameters ε_k are determined as follcws

$$\varepsilon_2 = 0, \qquad \varepsilon_3 = \varepsilon_0 R(u) \ , \qquad \varepsilon_4 = 0.125$$

where ε_0 is a positive parameter and R(u) is determined by equation (7).

3. APPROXIMATION OF THE EULER EQUATIONS

Consider the 1-D Euler equations

$$\frac{\partial U}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{8}$$

$$U = (\rho, \rho u, E)^T \ , \quad f = [\rho u, \rho u^2 + p, u(E+p)]^T$$

$$p = \frac{1}{\gamma M_\infty} \rho T, \quad E = \rho\left(C_v T + \frac{u^2}{2} \right)$$

The difference approximation used is

$$\left(I + \frac{\beta}{2} \frac{\Delta t}{\Delta x} \delta_x^- A^+ + \frac{\beta}{2} \frac{\Delta t}{\Delta x} \delta_x^+ A^-\right) \delta_t U_j^{n+1} = - \frac{\Delta t}{\Delta x} (F_j^+ + F_j^-) \tag{9}$$

$$F_j^\pm = \frac{\delta_x^\circ - 2\varepsilon_4 \delta_x^2}{1 + 2(\varepsilon_2^\pm - \varepsilon_4^\pm)\delta_x^\circ + (\frac{1}{6} - \varepsilon_3^\pm)\delta_x^2} f_j^\pm \tag{10}$$

[1] This work supported in part by the National Foundation of Natural Science

where A is the Jacobian matrix and $f^{\pm}=A^{\pm}U$, $A^{\pm}=S^{-1}\Lambda^{\pm}S$, and $\Lambda^{\pm}$ are the diagonal matrices with elements

$$\lambda^{\pm}_{k}=(\lambda_{k}+|\lambda_{k}|)/2$$

λ_k are the eigenvalues of the matrix A. In the computation the parameters $\varepsilon^{\pm}_{k}$ are determined as

$$\varepsilon^{\pm}_{2}=0, \quad \varepsilon^{\pm}_{3}=\pm\varepsilon_{0}R(\rho), \quad \varepsilon_{4}=\pm .125$$

4. NUMERICAL EXPERIMENTS

A. 1-D Shock Tube Problem

The upwind compact scheme was used to solve this problem. The computed results at the time t=0.14 are given in Fig.2.

B. 2-D Shock Reflection

The free Mach number is $M_{\infty}=2.9$, and the incident angle is 29 degree. Some computed results are given in Fig.3. The numerical experiments show

a. the third order accurate scheme (2) and (4) has nice solutions near the shock but still with small oscillations;

b. the third order accurate upwind compact scheme with dispersion regulator has high resolution of the shocks.

C. Supersonic Flow Around Blund Body

The scheme developed in ref.2 with artificial viscosity and with artificial dispersion was used to solve the axial symmetrical supersonic flow around a sphere-cone. The shock near the axis oscillates in the numerical solution obtained with artificial viscosity. When the scheme with artificial dispersion is used non oscillations are found in the numerical solutions.

REFERENCE

1. Ma Yanwen and Fu Dexun, AIAA paper No.87-1123.
2. Ma Yanwen and Fu Dexun, Lecture Notes in Physics Vol.264, 1986.

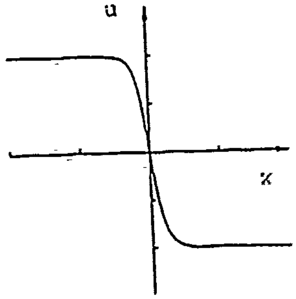
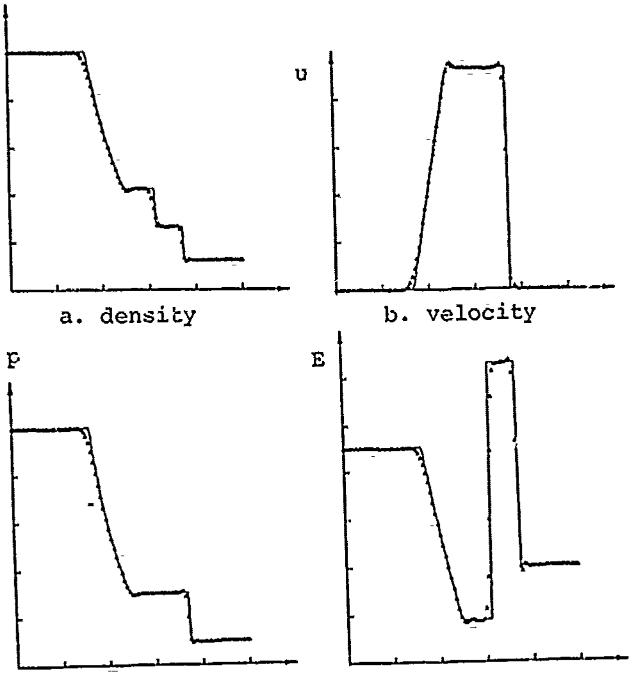


Fig.1 sketch of the shock



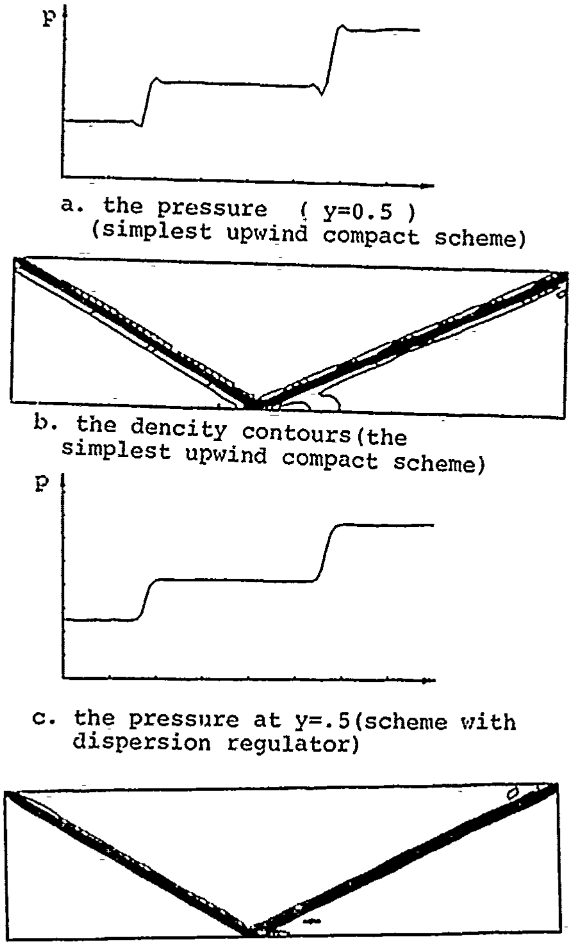
a. density

b. velocity

c. pressure

d. intenal energy

Fig.2 shock tube solutions at t=.14
(upwind compact scheme)



a. the pressure (y=0.5)
(simplest upwind compact scheme)

b. the dencity contours(the simplest upwind compact scheme)

c. the pressure at y=.5(scheme with dispersion regulator)

d. density contours(scheme with dispersion regulator)

NUMERICAL SIMULATION OF SUPERSONIC SEPARRATED FLOW OVER BLUNT CONE AT HIGH ANGLE OF ATTACK

Shen Qing Gao Shuchun and Zhang Hanxin
China Aerodynamics Research and Development Center
P. O. Box 211, Mianyang, Sichuan 621000, P. R. China

Abstract-Through a study for the properties of boundary layer equations, a space-marching technique for solving parabolized Navier-Stokes (PNS) equations has been developed by Zhang recently, and the axisymmetrical flows over blunt cones have been simulated successfully. In the present paper, the above technique is extented to the calculation of flow over blunt cone with cross-flow separation at high angle of attack •

Introduc.ion

Flow fields with cross-flow separations which dominat the shock layer during the reentry of configurations at high angles of attack can be simulated numerically by solving Navier Stokes(NS) equations, where Euler ones become invalid. Because the cost is very high and the computer storage is limited, the application of time dependent NS solver to the three-dimensional configrations is restricted greatly. Thus, discussion about various simplification for the NS equations has been made[1]. This analysis shows that the PNS equations are of the most interest. The reasons for this are (1) falling between the complete NS equations and the boundary layer equations, the set of equations are applicable to both inviscid and viscous flow regions, which means that the inviscid-viscous interaction is included automatically; (2) the PNS equations are hyperbolic parabolic type when certain conditions are met (i.e., the inviscid region of the flow is supersonic and the streamwise velocity component is positive), and then can be solved by advancing an initial plane of data in space. As a consequence, a substantial reduction in computation time and storage is achieved; (3) for stable marching downstream, the detail information can be supplied for the flow fields without reverse flow in the marching direction.

Recent years, various supersonic viscous flows have been computed with PNS method[2-10]. The key technique for stable computing is the proper treatment of the streamwise pressure gradient inside the subsonic region near body surface which produces the influence to the upstream and leads to the so called "departure solution". Numerous researchers[2-11] have done a lot of work to overcome this difficulty.

Zhang et al[11]. improved the PNS method based on the properties of the boundary layer equations. No departure solution occured with small marching step size. The computational results for the three-dimensional flow fields with the method of [11] are presented. The present PNS code has been used to calculate the laminar flow over shpere cone at angle of attack up to 20°. The computational results are compared with the data from references and experiments.

Numerical Method

Governing Equations

The conservative form of the PNS equations[3] is

$$E_\xi + F_\eta + G_\zeta = R_e^{-1} S_\zeta \tag{1}$$

where the perfect gas assumption is taken: $\gamma=1.4$, $P_r=0.72$, $T=\gamma M_\infty^2 p/\rho$ and the coefficient of viscousity (μ)is determined from Sutherland's law.

Numerical Algorithm

Eq. (1) can be rewrittenas

$$E_\xi^* + P_\xi + F_\eta + G_\zeta = R_e^{-1} S_\zeta \tag{2}$$

where $E^* = E - P$, and

$$P = J^{-1}\begin{bmatrix} 0 \\ (1-\omega)\xi_x p \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

ω is min $(1, kM_\xi^2)$, k is a safety factor and is taken as 0.8 in the calculation. For computing efficiently, an Euler implicit approximatly factored, finite difference algorithm in delta form is used, where P_ξ is treated explicitly. The difference equations of Eq. (2) are

$$\begin{aligned}
&[\tilde{A}^{**} + \Delta\xi(\delta_\eta \tilde{B}^*)]\Delta q^* = -(\tilde{A}^{**} - \tilde{A}^{**-1})q^* - \Delta\xi(\delta_\xi P \\
&\quad + \delta_\eta \tilde{F}^* + \delta_\zeta \tilde{G}^* - R_e^{-1}\delta_\zeta \tilde{S}^*) \\
&[\tilde{A}^{**} + \Delta\xi(\delta_\zeta \tilde{C}^* - R_e^{-1}\delta_\zeta \tilde{M}^*)]\Delta q^* = \tilde{A}^{**}\Delta q^* \\
&q^{*+1} = q^* + \Delta q^*
\end{aligned} \tag{3}$$

According to Zhang's technique, $\delta_\xi P^{(2)}$ in Eq. (3) is calculated by $(1-\omega)\cdot R_e^{-1}(S_\xi^{(2)})_\xi$. If Beam—Warming scheme is used, the fourth-order dissipation term of [3] is employed to supress high frequency oscillations. Eqs. (3) represent block tri-diagonal systems of equations which are solved using a matrix solver as described in [3]. It is important that the main diagonal elements occupy a dominant position in solving the matrix. Usually, implicit smoothing terms are used to enhance the main diagonal elements[9,10]. The same purpose is attained with limiting the marching step size, then no implicit smoothing terms are needed and higher accuracy is achieved. Also, the implicit NND scheme[12] has been used, and no obvious difference is found between results given by two methods separately.

Boundary Conditions

The body surface conditions (no-slip, a specified surface temperature, and zero normal pressure gradient) are implemented implicitly in the algorithm.

The out bow-shock is fitted using the "pressure approach" procedure. The pressure behind the shock is calculated from Eqs. (3) with one-side difference in the ζ-direction, and the remaining flow variables are evaluated by satisfying the Rankine-Hugoniot relations.

At symmetry plane, the reflect condition is employed implicitly.

Initial Conditions

Initial conditions for starting downstream marching computation are provided with the three-dimensional time-dependent NS code[13].

Results and Discussion

Caes 1

At first, the flow over a blunt sphere cone with 10° cone half—angle at no incident is calculated in order to demonstrate the efficiency of the algorithm. The flow conditions are

$M_\infty = 8$, $R_{eL} = 5 * 10^6 (L/R_n = 50)$,

$T_\infty = 72.46K$, $T_b = 200K$.

The results show good agreement with the data from [14] for the shock location in Fig. 1 and surface pressure in Fig. 2, which verifies the validaty of the present code.

Caes 2

Flow over a blunt sphere cone with 4.7° cone half—angle for an angle of attack of 20° is computed here. Flow conditions are

$M_\infty = 10$, $R_{eL} = 2.3 \times 10^6 (L/R_n = 10.52)$,

$T_\infty = 49K$, $T_b = 294K$.

Results are compared with those of [15]. Fig. 3a presents the shock shape of thepresent calculation on the pitch plane. Fig. 3b is an experimental photograph[8]. Fig. 4 and 5 present the surface pressureand heat transfer rate respectively. The comparison of calculated and measured[15] surface pressure variation with the circumferential angle at difference axial positions is presented in Fig. 6, and the similar comparison for the normalized heat-transfer rate is shown in Fig. 7. The comparison shows the good agreement between the computational and experimental results.

No adverse pressure gradient along the meridienal angle exists at x/L=0.11. However, a minimum pressure point develops off the lee meridian at x/L=0.445, which indicates the existance of the crossflow separation. As the result of the apparence of the secondary crossflow separation, thesecond minimum pressure point occurs between the first minimum point and the lee meridian at x/L=0.95. Fig. 8a presents the limiting streamlineswhere the development of the open separation is observed. The primary group of converged limiting streamlines and the second group of converged limiting streamlines manifest the primary and secondary separation lines. The experimental photograph (Fig. 8b) of [15]which provide us the real oilflow pattern shows the same image. The vortex structure is given by plotting crossflow velocity vectors in Fig. 9.

Conclusion

The space-marching technique developed by Zhang for the axisymmetrical flows over blunt cones is extented to simulate asymmetrical flow with crossflow separation. The results for blunt cones, specially the separated flow patterns calculated, are obtained satisfactorily. It is shown that the PNS method used in this paper is successful.

References

[1]. Anderson, D. A. , et al. , "Computational Fluid Mechanics and Heat Transfer",p. p. 247-251, 1984.
[2]. Vigneron, Y. C. , et al. , AIAA 78-1137, 1978.
[3]. Schiff, L. B. and Steger,J. L. , AIAA 79-130,1979.
[4]. Tannehill, J. C. and Venkatapathy, E. , AIAA 81-0049, 1981.
[5]. Chaussee, D. S. , et al. , AIAA 81-0050, 1981.
[6]. Rizk, Y. M. ,et al. , AIAA 81-1261, 1981.
[7]. Venkatapathy, E. , et al. , AIAA 82-0028, 1982.
[8]. Nicolet, W. E. , et al. , AIAA 82-0026, 1982.
[9]. Szema, K. Y. , et al. , AIAA 83-0211, 1983.
[10]. Srinivasan, G. R. , et al. , AIAA 84-0015, 1984.
[11]. Zhang, H. X. , et al. , A study of implicit space-marching method for the supersonic viscous flow over the blunt cone, "ACTA AERODYNAMICS SINICA" (in chinese),Vol. 8, No. 3, 1990.
[12]. Zhang. , H. X. ,"APPLICATION OF MATHEMA TICS AND MECHANICS"(in chinese),Vol. 12,No. 1,1991.
[13]. Shen, Q. , et al. , Applications of the NND scheme to the NS solution about the nose region of a space craft, "ACTA AERODYNAMICS SINICA" (in chinese), Vol. 7, No. 2, 1989.
[14]. Lyubimov, A. N. and Rusanov, V. V. , " Gas flows past blunt bodies", NASA TT-F715, Feb. , 1973.
[15]. Cérésuela, R. , et al. , AGARD CP-30, 1968.

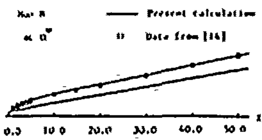


Fig.1 Axial-external shock shape.(M∞=8, α=0°)

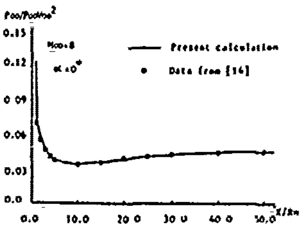
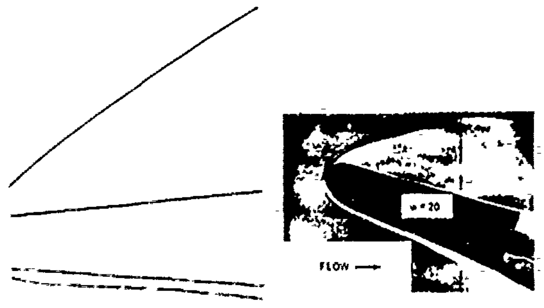


Fig.2 Axial surface pressure. (M∞=8, α=0°)



(a).Calculated (b).Experiment[15]

Fig.3 Shock shape on the pitch plane. (M∞=10, α=20°)

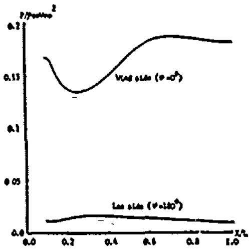


Fig.4 Axial surface pressure. (M∞=10, α=20°)

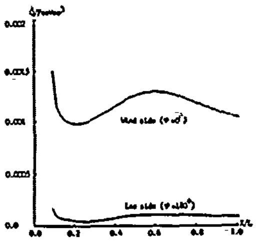


Fig.5 Axial surface heat transfer rate. (M∞=10, α=20°)

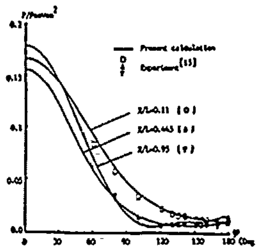


Fig.6 Circumferential surface pressure. (M∞=10, α=20°)

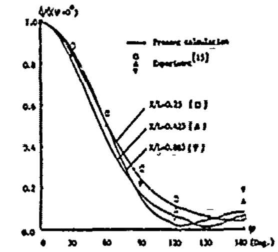
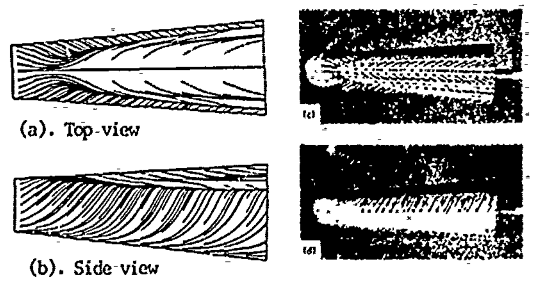


Fig.7 Circumferential heat transfer rate. (M∞=10, α=20°)



(a). Top-view

(b). Side view

Fig.8 Surface oilflow.

(a),(b) Calculated, (c),(d) Experiment[15]. (M∞ =10, α=20°)

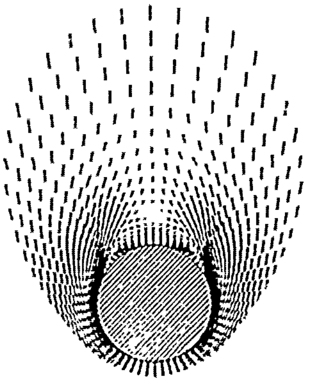


Fig.9 Cross-flow velocity vectors at X/L=0.988. (M∞=10, α=20°)

A FAST EFFICIENT ALGORITHM

FOR THREE DIMENSIONAL HYPERSONIC VISCOUS FLOW[1]

RU-QUAN WANG
Computing Center
Academia Sinica
Beijing, 100080, CHINA

and

JU-KUI XUE
Department of Physics
Northwestern Normal University
Lanzhou, 730070, CHINA

Abstract Based on the Single Level Conservative Supra-characteristic method (CSCM-S) proposed by Lombard et al. [1], we suggest a more fast efficient algorithm, which combines a single- marching technique for supersonic dominant zones with a multi-sweep procedure for complex flowfields. The new one requires about an order of magnitude less CPU time than the CSCM-S algorithm.

I. INTRODUCTION

Over the past ten years there has been a great development in efficient algorithms for comprissible flow. Among them the CSCM-S algoritm [1] is very attractive for multidimensional Euler and Navier-Stokes problems. The algorithm is based on an implicit symmetric Gauss-Seidel method along the flow direction together with an implicit block-tridiagonal diagonally dominant approximate factorization scheme along the other directions. In practice the global relaxation in whole computational domain is not economical for a large-scale supersonic dominant flowfield around a missile or space shuttle orbiter. In this paper we suggest the more fast efficient algorithm, which combines the single-sweep for the supersonic dominant flow with the multi-sweep process in the complex flow zones along the flow direction and adopts an explicit-implicit scheme rather than the full implicit scheme in the crossflow directions. We call such combinations as CSCM-S algorithm. Numerical experiments showed the efficiency of our new algorithm.

II. SIMPLIFIED NAVIER-STOKES EQUATIONS

The viscous flow is described by the full Navier-Stokes equations (NS). However, in many practical problems some viscous terms appearing in NS equations may be negligible. Based on this idea, the simplified Navier-Stokes (SNS) equations were derived using either the order of magnitude analysis or the viscous-inviscid interacting flow theory [2]. The conservative three-dimensional SNS equations have the following form in the generalized coordinates

$$J\frac{\partial q}{\partial t}+\frac{\partial E}{\partial \xi}+\frac{\partial F}{\partial \eta}+\frac{\partial G}{\partial \varsigma}=\frac{1}{Re}\left(\frac{\partial F_v}{\partial \eta}+\frac{\partial G_v}{\partial \varsigma}\right) \qquad (1)$$

where $q=(\rho,\rho u,\rho v,\rho w,e_t)^T$, $e_t=e+\frac{\rho}{2}\overline{V}^2$, $J=\frac{\partial(x,y,z)}{\partial(\xi,\eta,\varsigma)}$, $Re=\frac{\overline{\rho}_\infty\overline{U}_\infty L}{\overline{\mu}_\infty}$; E, F, G and F_v, G_v represent inviscid and viscous flux vectors, respectively. F_v, G_v contain such viscous terms that have orders of magnitude $O(1)$ and $O(R_e^{-\frac{1}{2}})$. The veiocity, density, temperature, viscosity and pressure were nondimensioned by $\overline{U}_\infty$, $\overline{\rho}_\infty$, $\overline{T}_\infty$, $\overline{\mu}_\infty$ and $\overline{\rho}_\infty\overline{U}_\infty^2$, respectively.

III. NUMERICAL METHOD

A. MULTIPLE SWEEP PROCEDURE

We consider an explicit-implicit difference equation for (1) at any grid point $(\xi_i,\eta_j,\varsigma_k)$

$$J\frac{\delta q^{n+1}}{\delta t}+\frac{\Delta E^{n+1}}{\Delta \xi}+\frac{\Delta F^{n+1}}{\Delta \eta}+\frac{\Delta G^{n+1}}{\Delta \varsigma}=\frac{1}{Re}\left(\frac{\Delta F_v^n}{\Delta \eta}+\frac{\Delta G_v^n}{\Delta \varsigma}\right) \qquad (2)$$

The CSCM flux-difference splitting [1] is chosen as the basic scheme of our algorithm, i.e. the inviscid flux difference can be expressed as

$$\begin{aligned}\Delta E&=\Delta E^{+}+\Delta E^{-}=\tilde{A}^{+}\nabla_\xi q+\tilde{A}^{-}\Delta_\xi q\\ \Delta F&=\Delta F^{+}+\Delta F^{-}=\tilde{B}^{+}\nabla_\eta q+\tilde{B}^{-}\Delta_\eta q\\ \Delta G&=\Delta G^{+}+\Delta G^{-}=\tilde{C}^{+}\nabla_\varsigma q+\tilde{C}^{-}\Delta_\varsigma q\end{aligned} \qquad (3)$$

where ∇q and Δq are the forward and backward differences, $A=\frac{\partial E}{\partial q}$, $B=\frac{\partial F}{\partial q}$, $C=\frac{\partial G}{\partial q}$ - Jacobian matrices and the matrices $\tilde{A}^{\pm}$, $\tilde{B}^{\pm}$, $\tilde{C}^{\pm}$ are obtained through A, B, C by the characteristic information along each coordinate direction and have the special form as

$$\tilde{A}^{\pm}=\overline{M}\overline{T}D^{\pm}\overline{T}^{-1}\overline{M}^{-1} \qquad \tilde{A}=\tilde{A}^{+}+\tilde{A}^{-}$$

where

$$\overline{M}=\begin{pmatrix}1&0&0&0&0\\ \overline{u}&1&0&0&0\\ \overline{v}&0&1&0&0\\ \overline{w}&0&0&1&0\\ \frac{\overline{u}^2+\overline{v}^2+\overline{w}^2}{2}&\overline{u}&\overline{v}&\overline{w}&1\end{pmatrix}$$

$$\overline{T}^{-1}=\begin{pmatrix}-\frac{1}{\overline{\rho}}&0&0&0&\frac{1}{\gamma\overline{P}}\\ 0&\overline{\overline{x_\eta}}&\overline{\overline{y_\eta}}&\overline{\overline{z_\eta}}&0\\ 0&\overline{\overline{x_\xi}}&\overline{\overline{y_\xi}}&\overline{\overline{z_\xi}}&0\\ 0&\overline{\overline{\xi}}_x/\overline{\rho c}&\overline{\overline{\xi}}_y/\overline{\rho c}&\overline{\overline{\xi}}_z/\overline{\rho c}&\frac{1}{\gamma\overline{P}}\\ 0&-\overline{\overline{\xi}}_x/\overline{\rho c}&-\overline{\overline{\xi}}_y/\overline{\rho c}&-\overline{\overline{\xi}}_z/\overline{\rho c}&\frac{1}{\gamma\overline{P}}\end{pmatrix}$$

$$P=p/(\gamma-1) \quad \overline{c}=\sqrt{\frac{\overline{\gamma p}}{\overline{\rho}}} \quad \overline{\overline{x_\eta}}=\frac{\overline{x_\eta}}{\sqrt{\overline{x_\eta}^2+\overline{y_\eta}^2+\overline{z_\eta}^2}}$$

$D^{\pm}=\frac{1}{2}(|\overline{\Lambda}|\pm\overline{\Lambda})/|\overline{\Lambda}|$, Λ is the diagonal matrix consisted of eigenvalues of the matrices $\overline{A}$. The M and T represent the transformed matrices from non-conservative to conservative equations and from characteristic to non-conservative ones, respectively. The bars indecate that the matrices contain averages of the adjacent grid point data. The viscous terms are approximated by central difference. Thus, the resulting difference scheme is

$$\begin{aligned}-\tilde{A}^{+}\delta q_{i-1,j,k}^{n+1}\quad -\tilde{B}^{+}\delta q_{i,j-1,k}^{n+1}+D\delta q_{i,j,k}^{n+1}&\\ +\tilde{A}^{-}\delta q_{i+1,j,k}^{n+1}+\tilde{B}^{-}\delta q_{i,j+1,k}^{n+1}&=-Lq_{i,j,k}^{n}\end{aligned} \qquad (4)$$

where $Lq_{i,j,k}^n=\tilde{A}^{+}\nabla_\xi q^n+\tilde{A}^{-}\Delta_\xi q^n+\tilde{B}^{+}\nabla_{eta}q^n+\tilde{B}^{-}\Delta_\eta q^n+\tilde{C}^{+}\nabla_\varsigma q^n+\tilde{C}^{-}\Delta_\varsigma q^n$, $D=I+\tilde{A}^{+}-\tilde{A}^{-}+\tilde{B}^{+}-\tilde{B}^{-}$. For solving eq.(4), the symmetric marching iteration is adopted along ξ-direction. i.e. the iteration consists of a forward-sweep step

$$(-\tilde{B}^{+},D,\tilde{B}^{-})\delta q^{*}=-L^{(n,*)}q \qquad (5)$$

$$q^{*}=q^{n}+\delta q^{*}, \qquad \delta q_{i+1,j,k}^{*}=0$$

and a backward-sweep step

$$(-\tilde{B}^{+},D,\tilde{B}^{-})\delta q^{n+1}=-L^{(*,n+1)}q \qquad (6)$$

$$q^{n+1}=q^{*}+\delta q^{n+1}, \qquad \delta q_{i-1,j,k}^{n+1}=0$$

where

$$L^{(n,*)}q=-Lq^n+\tilde{A}^{+}(q_{i,j,k}^n-q_{i-1,j,k}^n)+\tilde{A}^{-}(q_{i+1,j,k}^n-q_{i,j,k}^n)$$

$$L^{(*,n+1)}q=-Lq^{*}+\tilde{A}^{-}(q_{i+1,j,k}^{n+1}-q_{i,j,k}^{*})-\tilde{A}^{+}(q_{i,j,k}^{*}-q_{i-1,j,k}^{*})$$

B. SINGLE SWEEP TECHNIQUE

Note that the backward-sweep step (6) may be removed for the supersonic dominant flowfield and the numerical solution is found using only the forward-sweep (5). In fact, this can be easily done putting $\tilde{A}^{-}\equiv 0$ in eq.(4) and numerical results in this way are in good agreement with those obtained by the PNS space-marching technique. In this case the forward sweep step becomes the following form

$$(-\tilde{B}^{+},D,\tilde{B}^{-})\delta q^{n+1}=-L^{(n,n+1)}q \qquad (5')$$

$$q^{n+1}=q^{n}+\delta q^{n+1}$$

where

$$L^{(n,n+1)}q=-Lq^n+\tilde{A}^{+}(q_{i,j,k}^n-q_{i-1,j,k}^n)$$

[1] This work is supported by the National Natural Science Foundation of China.

$$I_{i}q^{n} = \widetilde{A}^{+}\nabla_{\xi}q^{n} + \widetilde{B}^{+}\nabla_{\eta}q^{n} + \widetilde{B}^{-}\Delta_{\eta}q^{n} + \widetilde{C}^{+}\nabla_{\zeta}q^{n} + \widetilde{C}^{-}\Delta_{\zeta}q^{n},$$

IV. NUMERIK AL TESTS

In order to test the efficiency of the CSCM-C algorithm, we computed some large-scale two and three dimensional hypersonic flows past a sphere-cone and a simple space shuttle orL'ter. A part of results for the sphere-cones is given here. In the paper [3] numerical test was accomplished for the sphere-cone hypersonic flow at $M_\infty = 20$ and zero angle of attack with the CSCM-S algorithm and the convergent solution was reached through 200 global sweeps. The same problem has been also computed by the CSCM-C under the following freestream conditions

$$M_\infty = 20, \quad Re = 0.6 \times 10^4, \quad T_\infty = 256k, \quad T_w = 1000k$$

Numerical solution was found by using the single sweep technique except for a small nose region, in which the steady solution was obtained through 100 global sweeps. Figs 1 and 2 show surface pressure and heating-rate distributions, respectively. They are in excellent agreement with those obtained by our 3-D VSL code [4]. The similar test was carried out for 15° sphere-cone three-dimensional flow at 20° angle of attack and the freestream conditions are the same as [5], i.e.

$$M_\infty = 10.6, \quad Re = 1.318 \times 10^5, \quad T\infty = 47.34k, \quad T_w = 300k$$

In this case the explicit-implicit scheme may reduce one-third of CPU time in comparison with the full implicit scheme (see table).

TABLE

COMPARISON OF DIFFERENT SCHEMES

AT ONE MARCHING STATION

SCHEME	EXPLICIT-IMPLICIT	IMPLICIT
CPU of per time step(sec.)	16.36	24.93
CPU of per grid point	0.021	0.032
total number of iteration	12	12
total CPU time	196.32	299.16

Figs. 3 and 4 represent axial distribution of surface pressures and heating rates along different meridianal planes. The points indecate experimental data. Clearly, the single-sweep technique is acceptable in the supersonic dominant flowfields.

V. CONCLUSION

As is well known that the time-dependent Navier-Stokes solver requires a large amount of CPU times to obtain a convergent numerical solution and it is more adequate to local complex flowfields. The space marching PNS algorithm can save an order of magnitude CPU time in comparison with the complete NS solver. However it has the inherent problem with stability, i.e. the exponentially growing solutions will occur if the streamwise pressure gradient is retained in the subsonic regions. The present CSCM-C algorithm consists of the space-marching technique and the time iteration and it is free of instability and requires less CPU time than the time-dependent Navier-Stokes solver. We expect the present algorithm would be more efficient to the large-scale flowfields around to the missile space shuttle orbiter configurations.

VI. REFERENCES

[1] C. K. Lombard et al., AIAA paper No.84-1533, 1984.

[2] Ru-quan Wang et al., Proc. of 3rd Symposium onv Computational Fluid Dynamics, Nagoya-Japan, 1989.

[3] J. Bardina et al., AIAA paper No. 85-0923, 1985.

[4] L. Q. Jiao et al., Acta Mechanica Sinica, No.3, 1980.

[5] J. W. Cleary, NASA TN-D-5450, 1969

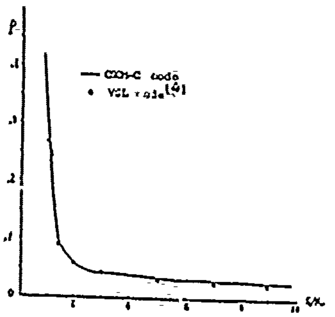


Fig.1. Surface pressure

along sphere-cone at zero angle of attack

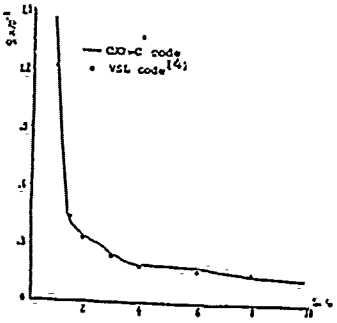


Fig.2. Surface heating-rate

along sphere-cone at zero angle of attack

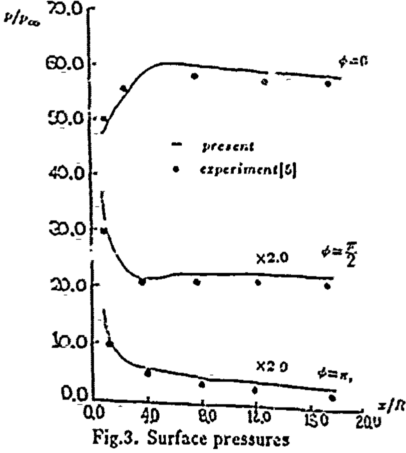


Fig.3. Surface pressures

along sphere-cone at 20° angle of attack

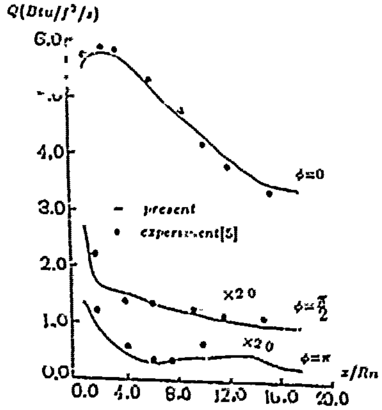


Fig.4. Surface heating-rates

along sphere-cone at 20° angle of attack

NUMERICAL SIMULATION FOR 3-D FLOWS USING NND SCHEMES

Ye Youda   Guo Zhiquan and Zhang Hanxin
China Aerodynamics Research and Development Center
P. O. Box 211, Mianyang, Sichang, 621000 P. R. China

Abstract : NND scheme which is developed by Zhang has been applied to the hypersonic flow around a shuttle—orbiter—like geometry proposed by CARDC. Euler equations in a finite difference are solved by using explicit space—marching algorithm and implicit marching iteration algorithm. Numerical calculations are performed for the problem under the conditions that Mach number is 7. 0 and the angle of attack is 5. 0 degree. Numerical results of these two algorithms are compared with each other and agree very well.

Introduction

In the calculation of complex flow fields containing shock waves, more attention has been paid to the shock capturing method. In order to capture shock waves smoothly without spurious oscillations near or in the shock regions, mixed dissipative schemes including free parameters with first order accuracy near shock and second order schemes elsewhere have been widely used [1-4]. There are inherent disadvantages in employing these schemes. Firstly, the free parameters are basically determined by numerical experiments. Secondly, the resolution of the shock is not very satisfactory. Naturally, the development of non—oscillatory dissipative schemes containing no free parameters with high resolution has much been emphasized recently, such as TVD schemes[5-9] and ENO schemes [10].

Through a study of the one dimensional Navier—Stokes equations, it was found that the spurious oscillations accuring near shock waves with finite difference equations are related to the dispersion term in the corresponding modified differential equations. If the sign of the dispersion coefficient is properly adjusted in order to make the sign change across the shock waves, the undesirable oscillations can be totally suppressed. Based on this discovery, the non—oscillatory and dissipative scheme containing no free parameters (NND scheme) is developed. This is one of " TVD ". In order to test the effectiveness of the scheme in space—marching calculations, we have carried out numerical simulations for two dimensional flows of shock and expension wave interaction flowfield. Moreover, the NND schemes have been employed for the calculation of hypersonic inviscid flow around the shuttle—orbiter—like geometry.

Numerical Algorithm

1, Semi—discreted NND scheme

For an one dimensional Euler equations :

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{1}$$

here U is a vector and F is a function of the vector U. The NND scheme of semi—discreted form is[11]:

$$\left(\frac{\partial U}{\partial t}\right)_j = -\frac{1}{\Delta x}(H^n_{j+1/2} - H^n_{j-1/2}) \tag{2}$$

where

$$H_{j+1/2} = F^+_{j+1/2L} + F^-_{j+1/2R}$$
$$F^+_{j+1/2L} = F^+_j + 1/2minmod(\Delta F^+_{j-1/2}, \Delta F^+_{j+1/2})$$
$$F^-_{j+1/2R} = F^-_{j+1} - 1/2minmod(\Delta F^-_{j+1/2}, \Delta F^-_{j+3/2})$$

2, Explicit NND scheme

For equation (1), two step predictor—corrector scheme is .

$$\begin{cases} \overline{U^{n+1}_j} = U^n_j - \frac{\Delta t}{\Delta x}(H^n_{j+1/2}, H^n_{j-1/2}) \\ U^{n+1}_j = \frac{1}{2}[U^n_j + \overline{U^{n+1}_j} - \frac{\Delta t}{\Delta x}(\overline{H^{n+1}_{j+1/2}}, \overline{H^{n+1}_{j-1/2}}) \end{cases} \tag{3}$$

This scheme is second order in time and space, and is TVD scheme. The maximun allowable Courant numeber is 1.

For simplicity, the predictor step of (3) can be also taken as :

$$\overline{U^{n+1}_j} = U^n_j - \frac{\Delta t}{\Delta x}(\Delta F^{+n}_{j-1/2} + \Delta F^{-n}_{j+1/2})$$

i. e. minmod(a, b)=0 in (3) is used.

3, Implicit NND scheme

For one dimensional Euler equation (1), we have

$$\left(\frac{\partial U}{\partial t}\right)^n_j = -\frac{1}{\Delta x}(H^n_{j+1/2} - H^n_{j-1/2}) \tag{4}$$

$$\frac{\partial U}{\partial t})^{n+1}_j = -\frac{1}{\Delta x}(H^{n+1}_{j+1/2} - H^{n+1}_{j-1/2}) \tag{5}$$

From (2), according to Crank—Nicolson method, U^{n+1}_j can be expressed as

$$U^{n+1}_j = U^n_j + \Delta t[(1-\theta)(\frac{\partial U}{\partial x})^n_j + \theta(\frac{\partial U}{\partial t})^{n+1}_j] \tag{6}$$

This expression is second order accuracy in time for θ=1/2 and is first order for θ=1. Substituting (4) and (5) into this expression, the following implicit scheme can be obtained :

$$U^{n+1}_j = U^n_j - \frac{\Delta t}{\Delta x}(\tilde{H}_{j+1/2} - \tilde{H}_{j-1/2}) \tag{7}$$

where

$$\tilde{H}_{j+1/2} - \tilde{H}_{j-1/2} = [(1-\theta)H^n_{j+1/2} + \theta H^{n+1}_{j+1/2}] - [(1-\theta)H^n_{j-1/2} + \theta H^{n+1}_{j-1/2}] \tag{8}$$

Define

$$k_1 = \frac{1}{2}minmod(\frac{\Delta F^+_{j+1/2}}{\Delta F^+_{j-1/2}}, 1)$$
$$k_2 = \frac{1}{2}minmod(\frac{\Delta F^-_{j+3/2}}{\Delta F^-_{j+1/2}}, 1)$$
$$k_3 = \frac{1}{2}minmod(1, \frac{\Delta F^+_{j-3/2}}{\Delta F^+_{j-1/2}})$$
$$k_4 = \frac{1}{2}minmod(1, \frac{\Delta F^-_{j-1/2}}{\Delta F^-_{j+1/2}})$$

The main idea in establishing our implicit scheme is to use the above limiters k_i at time step n. We have in general :

$$k^{n+1}_j \Delta F^{\pm n+1}_{j+1/2} = k^n_j \Delta F^{\pm n+1}_{j+1/2} + 0(\Delta t \Delta x)$$

Then equation (8) may be written as :

$$\tilde{H}_{j+1/2} - \tilde{H}_{j-1/2} = H^n_{j+1/2} - H^n_{j-1/2} + \theta(1 + k^n_1 - k^n_3) \cdot (\delta F^{+n+1}_j - \delta F^{+n+1}_{j-1}) + \theta(1 - k^n_2 + k^n_4) \cdot (\delta F^{-n+1}_{j+1} - \delta F^{-n+1}_j)$$

where

$$\delta F^{\pm n+1}_j = F^{\pm n+1}_j - F^{\pm n}_j$$

Since

$$\delta F^{+n+1}_j = A^{+n}_j \delta U^{n+1}_j + 0(\Delta t^2)$$
$$\delta F^{-n+1}_j = A^{-n}_j \delta U^{n+1}_j + 0(\Delta t^2)$$

where

$$\delta U^{n+1}_j = U^{n+1}_j - U^n_j$$
$$A^+ + A^- = \frac{\partial F}{\partial U} = A$$

The above equation (7) can be estimated by the equation

$$\tilde{A}^n_{j-1}\delta U^{n+1}_{j-1} + \tilde{B}^n_j \delta U^{n+1}_j + \tilde{C}^n_{j+1}\delta U^{n+1}_{j+1} = -\frac{\Delta t}{\Delta x}(H^n_{j+1/2} - H^n_{j-1/2}) \tag{9}$$

where

$$\tilde{A}^n_{j-1} = -\theta\frac{\Delta t}{\Delta x}(1 + k^n_1 - k^n_3)A^{+n}_{j-1}$$
$$\tilde{B}^n_j = I + \theta\frac{\Delta t}{\Delta x}(1 + k^n_1 - k^n_3)A^{+n}_j - \theta\frac{\Delta t}{\Delta x}(1 - k^n_2 + k^n_4)A^{-n}_j$$
$$\tilde{C}^n_{j+1} = \theta\frac{\Delta t}{\Delta x}(1 - k^n_2 + k^n_4)A^{-n}_{j+1}$$

This equation is what we have pursued in order to establish the so called implicit NND scheme, we can prove that this scheme for θ=1 is one of TVD schemes, and is unconditionally stable.

Numerical Results

1, Interaction of shock wave and expansion wave

In order to check out shock—capturing properties of NND schemes, we consider the problem showed in Fig. 1, OBAC is a curve, MN is a flat plate, FA is an oblique shock wave. Computational region is OABCDEFO. We specify a uniform Mach number 2. 9 at the left boundary FO, and the conditions at EF which is behind the shock, where the angle contained by the incident shock wave and the flat plate is 29°. A flow tangency condition is given at boundary OABC and MN, and the variables are extrapolated at boundary CED.

Since the flow is supersonic in this case, we can solve Euler equation by using the explicit space—marching algorithm.

The space—marching computation is started out at the left boundary, the space—marching step is 0. 01.

Fig. 2 shows the pressure contours evaluated by NND scheme, and the first order upwind (UP—1) and second order upwind (UP—2) schemes. It is quite clear that the spurious oscillations near or in the shock region can be totally suppressed with NND scheme, and NND scheme has high resolution to shock captuing.

2, Hypersonic Flow Around Shuttle=Orbiter—Like Geometry

We study the hypersonic flow around the shuttle—orbiter—like geometry, it is obvious that the flow is supersonic in the afterbody flowfield when the angle of attack is not very large, so the space—marching method is applied to solving the steady Euler equations. Through the transformation of coordinate system, we obtain the three—dimensional compressible steady Euler equations in dimensionless as :

$$\frac{\partial \tilde{E}}{\partial \xi}+\frac{\partial \tilde{F}}{\partial \eta}+\frac{\partial \tilde{G}}{\partial \zeta}=0 \qquad (10)$$

where ξ,η,ζ are the coordinates of streamwise wall—normal and circumferential directions respectively. Since equations (10) are hyprbolic and $\tilde{F}=\tilde{F}(\tilde{E})$; $\tilde{G}=\tilde{G}(\tilde{E})$ so the two step NND scheme (3) can be used directly.

When the angle of attack is large, the subsonic pockets will arise in the afterbody flowfied, so the time and space—marching methods should be applied to solving unsteady Euler equations. On the one hand, space—marching calculation is performed along streamwise direction, on the other hand, the time iteration is carried out on the every cross section. It is easy to obtain implicit marching—iteration scheme from model scheme (9), the Gauss—Seidel iteration will be used in the circumferential direction.

On the body surface, all variables can be determined by using tangential flow condition and four characteristic relations derived from Euler equations. At the shock wave, the Rankine—Hugoniot and one characteristic relation can be used to determine the shock wave shape and all variables. The initial profile can be given by time— dependent blunt body code [12]. The inviscid solution has been obtained on the following free stream condition : $M_\infty=7$, $\alpha=5°$. Fig. 3 gives the pressure contours on the body surface of leeward, the meridional surface $\varphi=90°$, $270°$, and out boundary cross section, the density contours are showed in Fig. 4, interaction of the body shock and wing shock are showed clearly on the meridional surface $\varphi=90°$, $270°$. Fig. 3 and Fig. 4 also show the contact discontinuties after the interaction of body shock and wing shock. The pressure distributions on meridional surface $\varphi=90°$ obtained with explicit and implicit scheme are compared in Fig. 5, which agree very well. The above results demonstrate that the capability of capturing shock and other contact discontinuties with NND schemes is satisfactory.

Concluding Remarks

According to the calculated results and the others [12–15], We are sure it is reliable to the NND scheme. The distinguished feature of the schemes is capability of capturing shock and other contact discontinuties, Futhermore, the schemes possess good stable characteristics and converged accuracy, which is essential to any high shock resolution scheme. In addition, the present form seems to be the simplest, meanwhile, the amount of the numerical work is much reduced in comparison with some other high resolution TVD schemes.

Reference

[1] Pullian, T. H. , AIAA 85—0438

[2] Jameson, A. and Yoon, S. , AIAA 85—0293

[3] Zhang, H. X. . et al, Applied Math. and Mech. (China), 4, 1, 1983

[4] Zhang, H. X. and Zheng, M. , Lecture Notes in Physics, 264, 1986

[5] Harten, A. , SIAM J. Anal. , 21, 1984

[6] Leer, B. V. , J. Comp. Phys. , 32, 1987

[7] Chakravarthy . S. R. and Osher, S. , AIAA 85—0363

[8] Davis, S. F. , ICASE Report 84—20, 1984

[9] Yee, H. C. , NASA TM—89464, 1987

[10] Harten, A. et al, ICASE Report 86—18, 1986

[11] Zhang, H. X. CARDC Report, 1988

[12] Ye, Y. D. , et al, Acta Aerodynamica Sinica, 7,3, 1989

[13] Zhang, H. X. , CARDC Report, 1989

[14] Zheng, M. and Zhang, H. X. , Acta Aerodynamica Sinica 7, 3, 1989

[15] Shen, T. et al, Acta Aerodynamica Sinica, 7,2, 1989

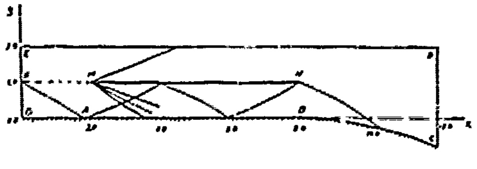


Fig.1 interaction of shock wave and expansion wave

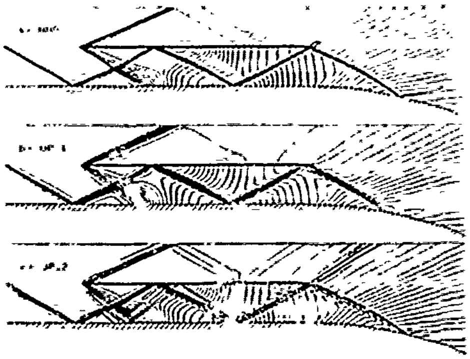


Fig.2 Pressure contours (M∞=2.9, Phi=29.0°)

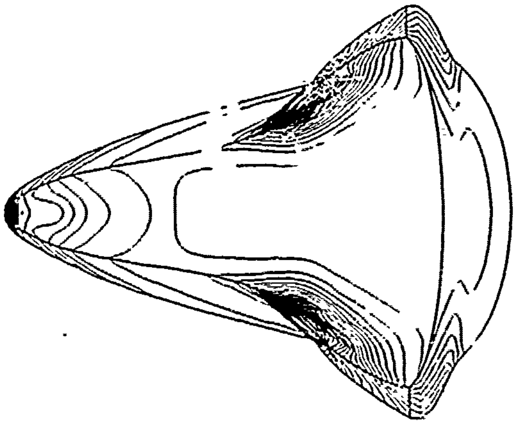


Fig.3 Pressure contours (M_∞=7,attack=5°,leeward)

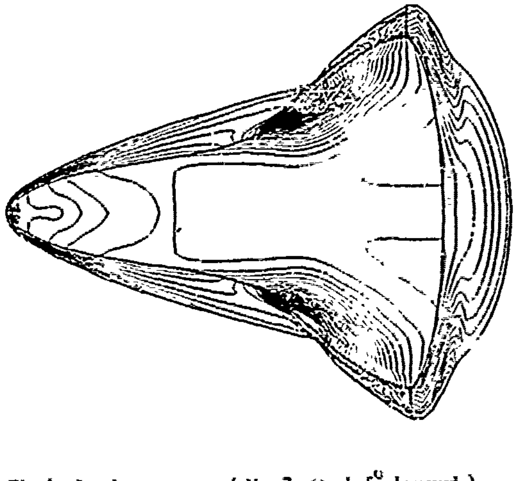


Fig.4 Density contours (M_∞=7,attack=5°,leeward)

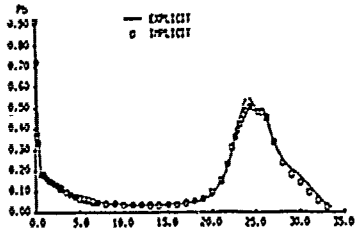


FIG. 5 PRESSURE DISTRIBUTION (M=7,ATTACK=5,PHI=90.0)

GRAVITATIONAL FORCES IN DUAL-POROSITY MODELS OF SINGLE PHASE FLOW*

Todd Arbogast
Department of Mathematical Sciences
Rice University
Houston, TX 77251-1892 U.S.A.

Abstract—A dual-porosity model is derived by the formal theory of homogenization. The model properly incorporates gravity in that it respects the equilibrium states of the medium.

1. Introduction

We consider flow in a naturally fractured reservoir which we idealize as a periodic medium as shown in Fig. 1. There are three distinct scales in this system, the pore scale, the scale of the average distance between fractures, and the scale of the entire reservoir. The concept of dual-porosity [4], [10] is used to average the two finer scales in such a way that the pore scale is recognized as being much smaller than the fracture spacing scale. The fracture system is modeled as a porous structure distinct from the porous structure of the rock (the *matrix*) itself.

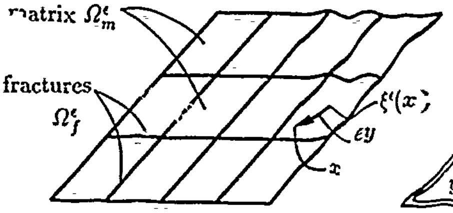


Fig. 1. The reservoir Ω.

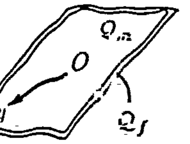


Fig. 2. The unit cell $\mathcal{Q}$.

Dual-porosity models can be derived by the technique of homogenization [2], [3], [6] (see also the general references [5], [7], and [9]). Briefly, we pose the correct microscopic equations of the flow in the reservoir and then let the block size shrink to zero. The resulting macroscopic model is formulated in six space dimensions, three of them represent the entire reservoir over which the fracture system flow occurs. At each point of the reservoir, there exists a three dimensional, "infinitely small" matrix block (surrounded by fractures) in which matrix flow occurs.

For single-phase, single component flow, it is recognized that diffusive, gravitational, and viscous forces affect the movement of fluids between the matrix and fracture systems; however, only diffusive forces are easily handled (see, e.g., [1], [4], [6], [8], [10], and the many multiphase models in the petroleum literature). Simply including gravity in the matrix of the standard model [2], [3], [6] creates an inconsistency in that when the fracture system is in gravitational equilibrium, the matrix system is not. In this paper we derive a consistent model.

2. The Microscopic and Macroscopic Models

Denote the reservoir by Ω. For a sequence of ϵ's decreasing to zero, we consider equivalent reservoirs with matrix blocks that are ϵ times the original size in any linear direction. Let Ω^ϵ_f and Ω^ϵ_m be the fracture and matrix parts of Ω, respectively. Each period of the reservoir is congruent to the unit cell $\epsilon\mathcal{Q}$, the period at point $x \in \Omega$ is denoted by $\mathcal{Q}^\epsilon(x)$. For the fixed unit cell $\mathcal{Q}$ (see Fig. 2), we write $\mathcal{Q}_f$ and $\mathcal{Q}_m$ for the fracture and matrix parts, respectively. Let the centroid of $\mathcal{Q}$ be the origin, and the centroid of $\mathcal{Q}^\epsilon(x)$ be $\xi^\epsilon(x)$. Then $x = \xi^\epsilon(x) + \epsilon y \sim x + \epsilon y$. Asymptotically, $x \sim \xi^\epsilon(x)$ selects a period and y specifies a point in the enlarged, congruent period $\mathcal{Q}$. Let ν denote the unit normal vector to the matrix fracture interface $\partial\Omega^\epsilon_m$ (or $\partial\mathcal{Q}_m$).

We use upper and lower case letters for fracture and matrix quantities, respectively. Let P (or p) be the fluid pressure, and Φ^* (or ϕ) and K^* (or k) be the porosity and permeability on the *pore* scale (so $\Phi^* \approx 1$ and K^* is very large). The fracture *system* porosity and permeability, Φ and K, are defined on the *fracture spacing* scale. Easily

$$\Phi = |\mathcal{Q}_f|\Phi^*/|\mathcal{Q}|, \tag{1}$$

where $|\ |$ denotes the volume of the set, while K is derived by homogenization. Finally, $\rho(P)$ (or $\rho(p)$) and μ are fluid density and viscosity, and g is the gravitational constant. Let e_j point in the jth Cartesian direction, where e_3 points down.

Define the function $\psi(x_3)$ as the solution to

$$\psi' = \rho(\psi)g; \quad \text{i.e.,} \quad \int_{\psi_0}^{\psi} \frac{d\pi}{\rho(\pi)} = g(x_3 - x_{3,0}). \tag{2}$$

Then $\nabla p - \rho(p)g\mathrm{e}_3 = 0$ if and only if $p = \psi(x_3 + \bar{x}_3)$ for some constant $\bar{x}_3$, and so $\psi(x_3 + \bar{x}_3)$ is the gravitational equilibrium pressure distribution. We note that the pseudopotential of the flow is given by $\psi^{-1}(p) - x_3$.

We ignore boundary conditions on $\partial\Omega$, external sources/sinks, and initial conditions since we are interested in internal flow.

The microscopic model: For the fracture flow,

$$\begin{aligned} &\Phi^*\frac{\partial}{\partial t}\rho(P^\epsilon) - \nabla\cdot\left[\mu^{-1}\rho(P^\epsilon)K^*\big(\nabla P^\epsilon - \rho(P^\epsilon)g\mathrm{e}_3\big)\right] \\ &\qquad = 0, \quad x \in \Omega^\epsilon_f, \end{aligned} \tag{3a}$$

$$\begin{aligned} &\mu^{-1}\rho(P^\epsilon)K^*\big(\nabla P^\epsilon - \rho(P^\epsilon)g\mathrm{e}_3\big)\cdot\nu \\ &\qquad = \epsilon\mu^{-1}\rho(p^\epsilon)k\big(\epsilon\nabla p^\epsilon - \rho(p^\epsilon)g\mathrm{e}_3\big)\cdot\nu, \quad x \in \partial\Omega^\epsilon_m. \end{aligned} \tag{3b}$$

For the matrix,

$$\begin{aligned} &\phi\frac{\partial}{\partial t}\rho(p^\epsilon) - \epsilon\nabla\cdot\left[\mu^{-1}\rho(p^\epsilon)k\big(\epsilon\nabla p^\epsilon - \rho(p^\epsilon)g\mathrm{e}_3\big)\right] \\ &\qquad = 0, \quad x \in \Omega^\epsilon_m, \end{aligned} \tag{4a}$$

$$\begin{aligned} &p^\epsilon = \psi\big(\psi^{-1}(P^\epsilon) + (\epsilon^{-1} - 1)(x_3 - \xi^\epsilon_3(x)) + \bar{\zeta}^\epsilon\big), \\ &\qquad\qquad x \in \partial\Omega^\epsilon_m. \end{aligned} \tag{4b}$$

On each $\mathcal{Q}^\epsilon(x)$, we need to define $\bar{\zeta}^\epsilon$. For a given P^ϵ, we can find for each constant $\bar{\zeta}^\epsilon$ the solution $\bar{p}^\epsilon$ of the steady-state problem corresponding to (4). So, for the given fracture pressure P^ϵ, we take the $\bar{\zeta}^\epsilon$ which gives rise to the $\bar{p}^\epsilon$ that satisfies

$$\int_{\mathcal{Q}^\epsilon_m(x)} \phi\rho(\bar{p}^\epsilon)\,dx = \int_{\mathcal{Q}^\epsilon_m(x)} \phi\rho(\bar{p}^\prime)\,dx, \tag{5}$$

where $\bar{p}^\prime$ is the steady state solution of the unscaled problem corresponding to (4), given by removing the two ϵ's appearing as coefficients in (4a) and replacing (4b) by $\bar{p}^\prime = P^\epsilon$. (In the case of an incompressible fluid, simply take $\bar{\zeta}^\epsilon = 0$.)

This ϵ family of microscopic models satisfies the following.

(i) Darcy flow governs the reservoir, and it does so in the standard way when $\epsilon = 1$ (since then $\bar{\zeta}^\epsilon = 0$);

(ii) For each ϵ, Darcy flow occurs in the fractures and within the *scaled* matrix blocks (i.e., if any matrix block $\mathcal{Q}^\epsilon_m$ is expanded to unit size $\mathcal{Q}_m$, the transformed equations

*This work was supported in part by the National Science Foundation and the State of Texas.

indicate that Darcy flow results);

(iii) If the fracture system is in gravitational equilibrium in the vicinity of a block, then the boundary conditions on that block reflect this gravitational equilibrium;

(iv) For fixed fracture conditions around any matrix block, the steady state matrix solution gives rise to the same mass as calculated from the steady-state solution of the unscaled matrix problem.

We require (iv) so that mass is conserved, since when we scale the matrix problem with (ii)-(iii), we change the pressures which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model [2], [3], [6] replaces (4b) with $p^\epsilon = P^\epsilon$, omits (5), and to be consistent needs to have $\rho(p^\epsilon)g$ replaced by $\epsilon\rho(p^\epsilon)g$ in (3b) and (4a). The novel expression (4b) can be viewed as a scaled continuity of pseudopotential, since we can rewrite it as

$$\psi^{-1}(p^\epsilon) - \left(\xi_3^\epsilon(x) + \epsilon^{-1}(x_3 - \xi_3^\epsilon(x)) + \bar{\zeta}^\epsilon\right) = \psi^{-1}(P^\epsilon) - x_3.$$

The macroscopic model: For the fracture flow,

$$\Phi\frac{\partial}{\partial t}\rho(P^0) + \frac{1}{|\mathcal{Q}|}\int_{\mathcal{Q}_m}\phi\frac{\partial}{\partial t}\rho(p^0)\,dy$$
$$-\nabla_x\cdot\left[\mu^{-1}\rho(P^0)K\left(\nabla_x P^0 - \rho(P^0)g\mathrm{e}_3\right)\right] = 0, \quad x\in\Omega, \quad (6)$$

where (1), (9), and (10) define the new coefficients. For the matrix flow, for each $x \in \Omega$,

$$\phi\frac{\partial}{\partial t}\rho(p^0) - \nabla_y\cdot\left[\mu^{-1}\rho(p^0)k\left(\nabla_y p^0 - \rho(p^0)g\mathrm{e}_3\right)\right]$$
$$= 0, \quad y\in\mathcal{Q}_m, \quad (7a)$$
$$p^0 = \psi\left(\psi^{-1}(P^0) + y_3 + \bar{\zeta}^0\right), \quad y\in\partial\mathcal{Q}_m, \quad (7b)$$

where ψ is defined by (2) and $\bar{\zeta}^0$ is defined by

$$\frac{1}{|\mathcal{Q}_m|}\int_{\mathcal{Q}_m}\phi\rho\left(\psi(\psi^{-1}(P^0) + y_3 + \bar{\zeta}^0)\right)dy = \phi\rho(P^0). \quad (8)$$

Note that no auxiliary steady-state problem need be solved.

The standard macroscopic model replaces (7b) by $p^0 = P^0$, omits (8), and should have $g = 0$ in (7a) to be consistent.

3. Formal Homogenization

We follow the homogenization of the standard model given in [2] and [6]. As usual, for some functions P^ℓ and p^ℓ, $\ell = 0, 1, 2, \ldots$, we assume the formal asymptotic expansions

$$x - \xi^\epsilon(x) \sim \epsilon y \quad\text{and}\quad \nabla \sim \epsilon^{-1}\nabla_y + \nabla_x,$$
$$P^\epsilon(x,t) \sim \sum_{\ell=0}^{\infty}\epsilon^\ell P^\ell(x,y,t), \quad x\in\Omega,\ y\in\mathcal{Q}_f,$$
$$p^\epsilon(x,t) \sim \sum_{\ell=0}^{\infty}\epsilon^\ell p^\ell(x,y,t), \quad x\in\Omega,\ y\in\mathcal{Q}_m,$$

where the P^ℓ are periodic in y with period $\mathcal{Q}_f$, reflecting the periodicity of the medium. We note that if some function F depends on $\pi^\epsilon \sim \sum_{\ell=0}^{\infty}\epsilon^\ell\pi^\ell$, then Taylor's Theorem shows that

$$F(\pi^\epsilon) \sim F\left(\sum_{\ell=0}^{\infty}\epsilon^\ell\pi^\ell\right) = F(\pi^0) + \sum_{\ell=1}^{\infty}\epsilon^\ell F^\ell,$$

for some F^ℓ that depend on the π^ℓ's.

Substituting the formal expansions into (3)-(5) and isolating the coefficients of powers of ϵ yield relations for the P^ℓ and p^ℓ.

We begin with two standard results which can be easily derived and appear in [2] and [6]. First, the ϵ^{-2} terms of (3a) and the ϵ^{-1} terms of (3b) imply that $P^0 = P^0(x,t)$ only. Second, the ϵ^{-1} terms of (3a) and the ϵ^0 terms of (3b) allow us to write

$$P^1 = \sum_{j=1}^{3}\frac{\partial P^0}{\partial x_j}\omega_j - \rho(P^0)g\omega_3 + \pi,$$

for some $\pi(x,t)$, where the $\omega_j(y)$, $j = 1,2,3$, are periodic across $\partial\mathcal{Q}$ and satisfy

$$-\nabla_y\cdot(\nabla_y\omega_j) = 0, \quad y\in\mathcal{Q}_f, \quad (9a)$$
$$\nabla_y\omega_j\cdot\nu = -\mathrm{e}_j\cdot\nu, \quad y\in\partial\mathcal{Q}_m. \quad (9b)$$

Recognizing that $(\epsilon^{-1} - 1)(x_3 - \xi_3^\epsilon(x)) \sim (1-\epsilon)y_3$, we have (7) from the ϵ^0 terms of (4).

We now consider (5). First, (4) or (7), without the time derivative term, implies $\bar{p}^0 = \psi\left(\psi^{-1}(P^0) + y_3 + \bar{\zeta}^0\right)$. For p^ϵ, the ϵ^{-2} terms of its defining equation and the ϵ^0 terms of its boundary condition imply $\bar{p}^0 = P^0$. Now a rescaling shows that

$$\int_{\mathcal{Q}_m^\epsilon(x)}\phi\rho(\bar{p}^\epsilon)\,dx \sim \int_{\mathcal{Q}_m}\phi\rho\left(\sum_{\ell=0}^{\infty}\epsilon^\ell\tilde{p}^\ell(x,y,t)\right)dy,$$

for some $\tilde{p}^\ell$ depending on the P^ℓ's and on $\bar{\zeta}^\epsilon$. A similar expression holds for the right side of (5), and so the ϵ^0 terms of (5) give the definition of $\bar{\zeta}^0$ as (8).

Finally, the ϵ^0 and ϵ^1 terms of (3a) and (3b) can be analyzed exactly as in the standard model [2], [6] to give (6), and the tensor K is seen to be given by

$$K_{ij} = \frac{K^*}{|\mathcal{Q}|}\left(\int_{\mathcal{Q}_f}\frac{\partial\omega_j}{\partial y_i}\,dy + |\mathcal{Q}_f|\delta_{ij}\right); \quad (10)$$

K is symmetric and positive definite (see, e.g., [3]).

References

1. T. Arbogast, *Analysis of the simulation of single phase flow through a naturally fractured reservoir*, SIAM J. Numer. Anal. 26 (1989), 12-29.
2. T. Arbogast, J. Douglas, Jr., and U. Hornung, *Modeling of naturally fractured reservoirs by formal homogenization techniques*, (to appear).
3. T. Arbogast, J. Douglas, Jr., and U. Hornung, *Derivation of the double porosity model of single phase flow via homogenization theory*, SIAM J. Math. Anal. 21 (1990), 823-836.
4. G. I. Barenblatt, Iu. P. Zheltov, and I. N. Kochina, *Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]*, Prikl. Mat. Mekh. 24 (1960), 852-864; J. Appl. Math. Mech. 24 (1960), 1286-1303.
5. A. Bensoussan, J. L. Lions, and G. Papanicolaou, "Asymptotic analysis for periodic structures," North-Holland, Amsterdam, 1978.
6. J. Douglas, Jr., and T. Arbogast, *Dual-porosity models for flow in naturally fractured reservoirs*, in "Dynamics of Fluids in Hierarchical Porous Formations," J. H. Cushman, ed., Academic Press, London, 1990, pp. 177-221.
7. H. I. Ene, *Application of the homogenization method to transport in porous media*, in "Dynamics of Fluids in Hierarchical Porous Formations," J. H. Cushman, ed., Academic Press, London, 1990, pp. 223-241.
8. H. Kazemi, *Pressure transient analysis of naturally fractured reservoirs with uniform fracture distribution*, Soc. Petroleum Engr. J. 9 (1969), 451-462.
9. E. Sanchez Palencia, "Non homogeneous Media and Vibration Theory," Springer-Verlag, Berlin, 1980.
10. J. E. Warren and P. J. Root, *The behavior of naturally fractured reservoirs*, Soc. Petroleum Engr. J. 3 (1963), 245-255.

MODELING OF THREE DIMENSIONAL UNSTABLE MISCIBLE DISPLACEMENT IN POROUS MEDIA

Mary Fanett Wheeler
Department of Mathematical Sciences
Rice University
Houston, TX 77251-1892 U.S.A.

and

Ashok Chilakapati and Clarence Miller
Department of Chemical Engineering
Rice University
Houston, TX 77251-1892 U.S.A.

Abstract Simulation of three dimensional unstable miscible displacement utilizing a numerical method which combines the modified method of characteristics is described. Numerical experiments are presented.

I. INTRODUCTION

In flow through a porous medium small flow disturbances are continuously generated due to heterogeneities of the medium, both physical and chemical. The processes of finger growth and interaction are linked to the heterogeneity of the medium and in particular to the spatial variation of porosity and permeability. For example viscous fingering arises in the displacement of a fluid in a porous medium by a less viscous fluid. That is small perturbations tend to grow with time producing large protrusions or fingers. A large number of miscible enhanced oil recovery processes are dominated by viscous fingering and cause a severe reduction in displacement efficiency. In groundwater fingering can cause uneven spread of a contaminant. Chemical heterogeneities arising from adsorption and ion exchange, further accentuate the instabilities.

Numerical simulation is a major tool in understanding the effects of various parameters on miscible viscous fingering. Investigation of the effect of the structure of the porous medium has been an active research area for many years. A major difficulty is in developing realistic descriptions of the medium for numerical simulation. Conditional simulation [4, 5, 7, 8, 1] involves the generation of synthetic porous media that are compatible with the available statistical information. To obtain a meaningful statistical result, many realizations must be carried out and then averaged. A number of statistical techniques that have been employed for generating realizations of a porous medium are described in [6]

In [7, 8] Moissis, Miller, and Wheeler studied by numerical simulation the effects of spatial variation in permeability and of the viscosity ratio on horizontal unstable miscible displacement. Their numerical experiments were limited to a two-dimensional porous media and to a linear (rectangular) geometry. In this paper we briefly discuss extensions of these results to three spatial dimensions. For a linear flood problem we formulate a numerical method combining a Galerkin characteristics method with continuous trilinear elements for the concentration of the invading fluid and a mixed finite element (cell-centered finite differences) for the pressure equation. Some numerical results are presented which involve a porous medium previously studied [3, 7].

II. METHODOLOGY

2.1 The Problem.

Consider incompressible miscible displacement in a porous medium having the shape of a rectangular parallelpiped. The pore space of the medium is initally filled with the resident fluid. The medium is flooded at the side $x = 0$ with pure invading (or displacing) fluid. Thus x is the principal flow direction and y and z are the directions transverse to the flow. Let Lx, Ly, and Lz be the x, y, and z dimensions of the parallelpiped and let $\Omega = (0, Lx) \times (0, Ly) \times (0, Lz)$. Under the above assumptions, the displacement can be modelled by the following set of equations.

$$\mathrm{u} = -\frac{k}{\mu}\nabla p \tag{1}$$

$$\nabla \cdot \mathrm{u} = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\phi c) + \mathrm{u} \cdot \nabla c = \nabla \cdot D \nabla c \tag{3}$$

Equations (1) (2) can be combined to yield the pressure equation and (3) is the concentration equation. Here D is the diffusion dispersion tensor. The eigenvectors of this tensor are orthogonal with one of the eigenvectors being u and the eigenvalues α_l, α_t and α_t.

We impose the Dirichlet boundary conditions $p = p_0(t)$ and $c = c_0(t)$ at the inflow boundary $x = 0$. The outflow boundary conditions at $x = Lx$ are

$$p = 0 \tag{4}$$

$$\frac{\partial c}{\partial t} + \mathrm{u} \cdot \nabla c = 0 \tag{5}$$

The remaining faces have noflow boundary conditions. The initial condition is $c = 0$. The viscosity μ in (1) is a function of the concentration c of the invading fluid. The equation of state used in this work is

$$\frac{\mu}{\mu_r} = \left[1 + (M^{1/4} - 1)c\right]^{-4} \tag{6}$$

where μ_r and μ_i are the viscosities of the resident fluid and the invading fluid respectively and $M = \mu_r/\mu_i$.

2.2 Numerical Method

The system of coupled equations is numerically approximated by a predictor corrector time stepping method which combines the modified method of characteristics for the concentration equation and a mixed finite element method (cell centered finite difference method) for the pressure equation (MMOC-MFE).

Let $\mathcal{M}_h(Z_h)$ denote the finite dimensional space spanned by the continuous (discontinuous) trilinear (constants) defined on Ω. Let $V_h = V_h(x) \times V_h'(y) \times V_h'(z)$ where $V_h(x)$ denotes the tensor product space of continuous piecewise linears on $[0, Lx]$ and discontinuous piecewise constants on $[0, Ly] \times [0, Lz]$. $V_h(y)$ and $V_h(z)$ are defined similarly.

Let

$$V_h'(z) = V_h(z) \cap \{v | v(x,y,0) = v(x,y,Lz) = 0.\} \qquad (7)$$

Similarly we define $V_h'(y)$.

We define the inner product $< v, u >= \int_\Omega vu dx dy dz$. Let $\Delta t^n > 0$ and $t^n = n\Delta t$.

In the MMOC-MFE formulation we seek an approximation $(C; P, \mathrm{U})$ in $\mathcal{M}_h \times Z_h \times V_h$ to the solution $(c; p, \mathrm{u})$ as follows.

For $n \geq 0$, the MFE approximation (P^n, U^n) is defined by:

$$< \frac{\mu(C^{n,0})}{k}\mathrm{U}^n, \mathrm{v} > - < P^n, \nabla \cdot \mathrm{v} >$$
$$= \int_0^{L_z} \int_0^{L_y} p_0(y,z,t^n) v(0,y,z) dy dz, \qquad \mathrm{v} \in V_h \quad (8)$$
$$< \nabla \cdot \mathrm{U}^n, w >= 0, \qquad w \in Z_h \qquad (9)$$

where $C^{0,0} = C^0$.

For $n \geq 1$ the MMOC approximation $C^{n,k}, \quad k = 0, 1$, is defined by

$$< \phi \frac{C^{n,k} - \bar{C}^{n-1}}{\Delta t}, \chi > + < \mathrm{D}(\mathrm{U}^{n-1+k}) \nabla C^n, \nabla \chi >= 0, \qquad (10)$$

for $\chi \in \mathcal{M}_h$ where

$$C^{n,k}(0,y,z) = 1 \qquad (11)$$
$$C^{n,k}(1,y,z) = \bar{C}^{n-1}(1,y,z), \qquad (12)$$
$$(13)$$

$(y,z) \in (0,Ly) \times (0,Lz)$. Here

$$\bar{C}^{n-1}(\mathrm{x}) = C^{n-1}\left(\mathrm{x} - \frac{E\bar{\mathrm{U}}^n(\mathrm{x})}{\phi}\Delta t\right) \qquad (14)$$

and

$$C^n = C^{n,1} \qquad (15)$$

$E\bar{\mathrm{U}}^n(\mathrm{x})$ is an approximate average velocity between the times t^{n-1} and t^n, which is computed by segmenting the time step Δt into smaller sub-time steps and using a predictor-corrector procedure to determine the velocity along the characteristic in each sub-time step.

The combination of the MFEM and the MMOC has been previously applied to the solution of miscible displacement problems in two spatial variables [9, 2, 7, 8].

III. NUMERICAL RESULTS

The following data has been used in the simulation. $\alpha_l = 2.76 \times 10^{-3}$, $\alpha_t = 7.73 \times 10^{-5}$ and zero molecular diffusion. The inflow boundary condition for c was unity. We set

$$Lx = Ly = Lz = 1.$$

The isotropic permeability used in the model is described graphically in Fig.1 for a high and a low permeability planes. Corresponding concentration profiles are shown in Fig.2.

[1] M. S. Bartlett. *The Statistical Analysis of Spatial Pattern.* Chapman and Hall, London, 1975.

[2] R. E. Ewing, T. F. Russell, and M. F. Wheeler. Simulation of miscible displacement using mixed methods and a modified method of characteristics. *SPE 12241*, 1983.

[3] R. M. Giordano, S. J. Salter, and K. K. Mohanty. The effects of permeability variations on flow in porous media. *SPE 14365*, 198.

[4] A. B. Journel and Ch. J. Hujibregts. *Mining Geostatistics.* Academic, New York, 1978.

[5] L. W. Lake. *A Marriage of Geology and Reservoir Engineering*, volume 11, pages 177–198. Springer-Verlag, Berlin, 1987.

[6] A. Mantoglou and J. W. Wilson. The turning bands method for simulation of random fields using line generation by a spectral method. *Water. Resour. Res.*, 18:1379–1394, 1982.

[7] D. Moissis, C. A. Miller, and M. F. Wheeler. *A Parametric Study of Viscous Fingering*, volume 11, pages 227–247. Springer-Verlag, Berlin, 1987.

[8] D. E. Moissis and M. F. Wheeler. pages 243–270. Academic, New York, 1990.

[9] T. F. Russell, M. F. Wheeler, and C. Chiang. *Large Scale Simulation of Miscible Displacement by Mixed and Characteristic Finite Element Methods*, pages 85–107. SIAM, Philedelphia, 1986.

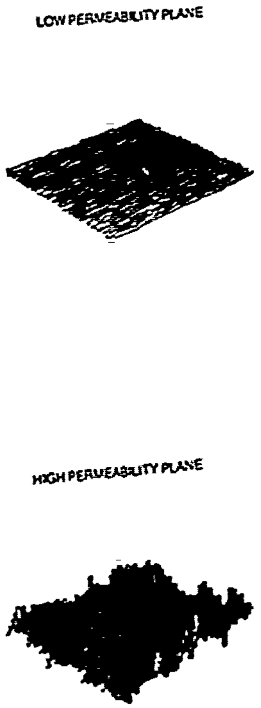


Fig. 1

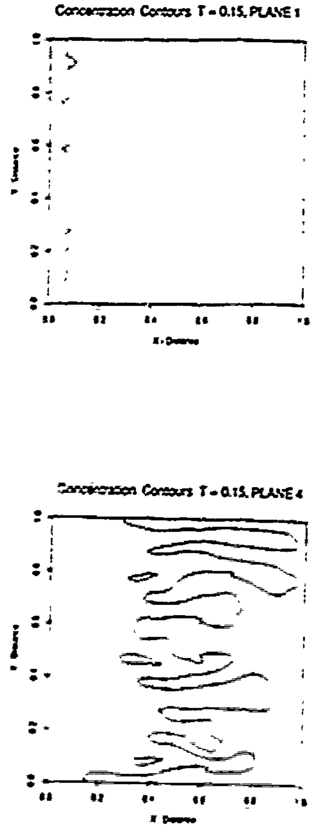


Fig. 2, M=10

ORDER OF CONVERGENCE ESTIMATES FOR FINITE ELEMENT APPROXIMATIONS OF DEGENERATE PARABOLIC SYSTEMS MODELLING REACTIVE SOLUTE TRANSPORT IN POROUS MEDIA

PETER KNABNER
Universität Augsburg
Institut für Mathematik
Universitätsstr. 8
D-8900 Augsburg, Germany

Abstract - The semidiscrete finite element approximation is studied for a semilinear reaction-diffusion system consisting of a parabolic and an ordinary differential equation, where the nonlinearity is non-Lipschitz, and a related pde of 'porous medium equation' type. Based on stability estimates for the continuous problem, order of convergence estimates are proved for linear elements in energy- and L^2-norms, which partially are optimal.

We consider the finite element approximation of equations, which are conceivable as a macroscopic *model for transport and adsorption in porous media* (cf. [4] for details). A water flow regime, characterized by the *water content* Θ and the *flux* vector field q and assumed to be known, causes the transport of a *solute with concentration* u by *convection* and *diffusion/dispersion*. The substance undergoes a *surface reaction with the porous sceleton* like adsorption, i.e. there is an *adsorbed concentration* v. The adsorption reaction may be either in non equilibrium, leading to

$$\begin{aligned} &\partial_t(\Theta u) + \rho\partial_t v - \nabla\cdot(D\nabla u - qu) = 0, \\ &\partial_t v = k(\varphi(u) - v) \quad \text{in } Q_T := \Omega\times(0,T], \end{aligned} \qquad (KA)$$

or in equilibrium, which gives rise to $v = \varphi(u)$, i.e.

$$\partial_t(\Theta u) + \rho\partial_t\varphi(u) - \nabla\cdot(D\nabla u - qu) = 0 \quad \text{in } Q_T. \qquad (EA)$$

(KA) or (EA) are supplemented by boundary and initial conditions

$$\begin{aligned} &(D\nabla u - qu)\cdot n = F\,(\geq 0) \quad \text{on } S_{1T} := S_1\times(0,T], \\ &D\nabla u\cdot n = 0 \quad \text{on } S_{2T} := S_2\times(0,T], \\ &u(.,0) = u_0\,(\geq 0)\;[v(.,0) = v_0\,(\geq 0)] \quad \text{in } \Omega. \end{aligned} \qquad (1)$$

Hereby $\Omega\subset\mathbb{R}^N$ is a bounded domain, $\partial\Omega = S_1\dot{\cup}S_2$, such that $q\cdot n \leq 0$ on S_1 and $q\cdot n \geq 0$ on S_2, n being the outward normal. The nonlinearity φ, the *adsorption isotherm*, only fulfills

$$\begin{aligned} &\varphi\in C^{0,p}[0,\infty)\cap C^{0,1}_{loc}(0,\infty) \text{ for some } p\in(0,1), \\ &\varphi(0) = 0,\ \varphi(s) > 0 \text{ for } s > 0,\ \varphi \text{ is non-decreasing,} \end{aligned} \qquad (2)$$

such that degeneration may occur at $u = 0$ leading to finite speed of propagation of supp u (and supp v) and limited smoothness of the solution. For the following results certain conditions are necessary with respect to the coefficients. These are fulfilled if the *rate parameter* $k > 0$ in (KA) is a constant and the other coefficients depend only on space satisfying

$$\begin{aligned} &\Theta(x)\geq\Theta_0 > 0,\quad \rho(x)\geq\rho_0 > 0,\quad \nabla\cdot q(x) = 0, \\ &D(x) \text{ is symmetric and positive definite uniformly in } x. \end{aligned} \qquad (3)$$

One may think of other rate functions, i.e. descriptions of $\partial_t v$, than the explicit one in (KA), but this specific structure is important for the following considerations.
For the semilinear case (KA) we have despite of the possible degeneration the optimal Lipschitz stability of u with respect to the energy norm

$$|u|_{Q_T} := \max_t \|u(.,t)\|_{L^2(\Omega)} + \|\nabla u\|_{L^2(Q_T)}.$$

Theorem 1. *Let (u_i, v_i) be weak solutions for the data u_{0i}, v_{0i} and F_i, then*

$$|u_1 - u_2|_{Q_T} \leq C(\|u_{01} - u_{02}\|_{L_2(\Omega)} + \|v_{01} - v_{02}\|_{L_2(\Omega)} + \|F_1 - F_2\|_{L^2(S_{1T})}).$$

□

Here and in the following $C > 0$ is a constant independent of the quantities, with which it is multiplied. The *proof* (cf. [2], [4]) consists of three basic steps:

(I) Test the pde with the primitive of $u := u_1 - u_2$,

(II) Test the ode with u,

(III) Test the pde with u.

These steps will reappear in the proofs of all the following assertions.

For $k\to\infty$ we expect convergence of (KA) to (EA). Under certain conditions on the data the speed can be estimated (cf. [4] for details of the convergence proof):

Theorem 2. *Let $(u^{(k)}, v^{(k)})$ be weak solutions of (KA) for the rate parameter k and let u be the weak solution of (EA) for the same data, then:*

$$\|u^{(k)} - u\|_{L^2(Q_T)} \leq C\left(\frac{1}{k}\right)^{1/2}.$$

□

This justifies the *kinetic approximation for (EA)*, i.e. to approximate the solution of (EA) by the solution of (KA) for large k.

We now turn to the finite element approximation of (KA), where we only consider the discretization in space. Let $\Omega\subseteq\mathbb{R}^2$ be a bounded domain with smooth $\partial\Omega$ and T_h a regular triangulation of Ω. The discretization parameter h is given by the maximal diameter of the triangles $T\in T_h$. To simplify the notation, we ignore $\Omega\,\Delta\cup_{T\in T_h} T$, i.e. we treat Ω as a polygonal domain. Let ρ be a constant and define

$$S_h := \{\chi \in C(\bar{\Omega}) \mid \chi_{|T} \text{ is linear for each } T \in \mathcal{T}_h\},$$
$$(f,g) := \int_\Omega f\,g\,dx. \tag{4}$$

Then the *consistent semidiscrete Galerkin approximation* is given by $u_h, v_h : [0;T] \to S_h$ satisfying

$$(\partial_t(\Theta u_h), \eta) + \rho(\partial_t v_h, \eta) - (L_h(u_h), \eta) = 0, \tag{5}$$

$$(\partial_t v_h, \eta) = (k(\varphi(u_h) - v_h), \eta) \quad \text{for} \quad \eta \in S_h,\ t \in (0,T], \tag{6}$$

$$u_h(0) = u_{0h}, \quad v_h(0) = v_{0h}, \tag{7}$$

where $L_h = L_h(t)$ is defined as follows: For $\eta \in S_h$:

$$(L_h(u), \eta) := -(D\nabla u - qu, \nabla\eta) - \int_{S_2} q \cdot n\, u\eta\, d\sigma + \int_{S_1} F\eta\, d\sigma.$$

The data are assumed to be sufficiently smooth and u_{0h}, v_{0h} are taken as the L^2-projections onto S_h.
Despite of the missing Lipschitz continuity of φ there is an optimal order of convergence estimate for the energy norm:

Theorem 3. *Let (u,v) be the weak solution of (KA) and (u_h, v_h) the consistent semidiscrete Galerkin approximation, then.*

i) $|u - u_h|_{Q_T} \le Ch$,

ii) $\|u - u_h\|_{L^2(Q_T)} \le Ch^{\frac{2}{2-p}}$.

Hereby p is the Hölderexponent of φ.

□

The *proof* uses an auxiliary linear problem by freezing the nonlinearity at the solution. Let (u_h^*, v_h^*) denote the Galerkin approximation for this problem. Then optimal convergence results for $u - u_h^*$ are well-known. The remainder $u_h^* - u_h$ is investigated by means of the basic steps in the proof of Theorem 1 and reasonings in [1]. It is open, whether also in $\|.\|_{L^2(Q_T)}$ the optimal estimates $O(h^2)$ holds. Investigations of uniform convergence are in progress.
The consistent approximation is only asymptotically in accordance with the physical picture insofar (6) is not a local equation in space. This is achieved by considering the *semidiscrete Galerkin approximation with mass lumping*, which is defined by (7) and

$$(\partial_t(\Theta u_h), \eta) + \rho(\partial_t v_h, \eta)_h - (L_h(u_h), \eta) = 0, \tag{8}$$
$$(\partial_t v_h, \eta)_h = (k(\varphi(u_h) - v), \eta)_h \quad \text{for} \quad \eta \in S_h,\ t \in (0,T], \tag{9}$$

where

$$(f,g)_h := \int_\Omega I(f,g)dx \quad \text{and for} \quad u \in C(\bar{\Omega})$$
$$I(u) \in S_h \quad \text{is defined by} \quad I(u)(P) = u(P) \quad \text{for all nodes } P \text{ of } \mathcal{T}.$$

In fact (9) is equivalent with the collocation approach requiring the ode to be fulfilled at the nodes P. Proceeding as in the proof of Theorem 3 we are lead to an additional term which can be interpreted as the quadrature error. If the solution u of (KA) would have the *nondegeneracy property* with $\alpha = \frac{2}{1-p}$, which roughly states that $u^{1/\alpha}$ grows linearly away from ∂ supp u into supp u, then the estimates of Theorem 3 can be shown along the lines of [1]. We conjecture that this property holds true, as this is the case for travelling wave solutions (cf. [3]), but there is no proof up till now. Without nondegeneracy property it is possible to consider (7), (8) for a regularized φ_ε and to adapt the regularization parameter ε to h.
This leads again to an order of convergence estimate, but weaker than in Theorem 3.

Finally, we consider the convergence of the *semidiscrete kinetic approximation.* A combination of Theorem 2 and an estimate similar to Theorem 1, but uniform in the rate parameter k, leads to

Theorem 4. *Let u be the weak solution of (EA), $(u_h^{(k)}, v_h^{(k)})$ the solutions of (5), (6), (7) for the rate parameter k, then:*

$$\|u - u_h^{(k)}\|_{L^2(Q_T)} \le C\, h^{\frac{1}{2-p}}$$

for $k = O(h^{-\frac{2}{2-p}})$.

□

References

[1] Barrett, J., Shanahan, R.. *Finite Element Approximation of a Model Reaction-Diffusion Problem with a Non-Lipschitz Nonlinearity.* Preprint (submitted).

[2] van Duijn, C. J., Knabner, P.: *Solute Transport through Porous Media with Slow Adsorption.* In: "Free Boundary Problems: Theory and Applications", Vol. I (K.-H. Hoffmann, J. Sprekels, eds.), Pitman Research Notes in Mathematics 185 (1990), 375-388.

[3] van Duijn, C. J., Knabner, P.. *Solute Transport in Porous Media with Equilibrium and Non-equilibrium Multiple-Site Adsorption. Travelling Waves.* J. reine angew. Math. (1991) (in press).

[4] Knabner, P.: *Mathematische Modelle für Transport und Sorption gelöster Stoffe in porösen Medien (in German).* Verlag P. Lang, Frankfurt/M., 1991 (in press).

COMPUTATIONAL ASPECTS OF CONTAMINANT TRANSPORT WITH NONLINEAR ADSORPTION

CLINT DAWSON
Department of Mathematical Sciences
Rice University
Houston, TX 77251-1892 U.S.A.

Abstract Simulation of advection-diffusion equations modeling solute tranport in groundwater by higher order Godunov-mixed [illegible] element techniques is described. In this approach, a higher order [illegible] method models advection, while the mixed method models diffus[illegible] These methods are especially useful for problems with nonlinearities, [illegible] as nonlinear adsorption terms. Numerical results for solute transport with instantaneous, nonlinear adsorption are presented

I. INTRODUCTION

As noted in two recent reports by the United States Environmental Protection Agency [1] and Department of Energy [2], the potability of groundwater at many locations in the United States is being adversely affected by the introduction of hazardous chemicals into the subsurface Such chemicals include BTX (benzene, toluene, xylene), gasoline, herbicides, and pesticides Modeling the flow of these types of chemicals through the subsurface has seen increasing interest in recent years.

The flow of contaminants in groundwater is influenced by many factors, including the aquifer characteristics (hydraulic conductivity, porosity, etc), the presence of microorganisms capable of biodegrading certain compounds, and the chemical process of adsorption. Biodegradation is an important aspect of groundwater flow, as many compounds may be eliminated by natural biodegradation. Moreover, natural biodegradation processes may be enhanced to effectively remove contaminants [1]. Adsorption, which is a retardation/release reaction between the solute and the surface of the porous structure, is also a significant factor in contaminant movement. Adsorption can have the effects of segragating a hazardous compound from the groundwater, and slowing the overall movement of the chemical species.

The author and M. F. Wheeler have developed and tested a numerical algorithm for modeling multidimensional, multicomponent contaminant flow which includes the effects of biodegradation and linear adsorption, see for example [3, 4]. In this paper, we consider one-dimensional flow of a chemical species in groundwater undergoing (possibly) nonlinear adsorption. In an earlier paper [5], the author and M. F. Wheeler described a first-order method for simulating this problem. We will describe here a higher-order extension of this technique and use it to study nonlinear adsorption phenomena in one space dimension.

II. THE MATHEMATICAL MODEL

Let c denote the concentration of a chemical species in solution We assume a source of solute $c = c_0$ at $x = 0$, and assume flow is in the positive x direction. In one space dimension, conservation of mass yields

$$\phi c_t + \rho A_t + uc_x - Dc_{xx} = 0, \quad x > 0, \ t > 0, \tag{1}$$

$$c(x,0) = c^0(x). \tag{2}$$

Here, the positive constants ϕ, u (cm/h), and D/ϕ (cm^2/h) denote porosity, Darcy velocity, and the sum of the molecular diffusion and mechanical dispersion coefficients, respectively. The term ρA represents the amount of solute adsorbed, where ρ (g/cm^3) is the bulk density In many cases of physical interest, flow is advection-dominated; that is, u is much larger than D multiplied by some appropriate length scale. This causes numerical difficulties, which we will discuss in more detail below.

The term ρA is in general heterogeneous, depending on the adsorbent surfaces. The adsorption process can be divided into two classes, equilibrium and non-equilibrium. We will only consider the case of equilibrium adsorption. Adsorption is assumed to be in equilibrium when the reaction kinetics occur at a much faster rate than the rate of transport. In this case, A can be written as (see [6, 7]) $A = \Psi(c)$, where $\Psi(c)$ is an adsorption isotherm. We will assume $\Psi(c)$ is described by the Freundlich isotherm,

$$\Psi(c) = K_d c^p, \quad 0 < p \leq 1, \tag{3}$$

where K_d (cm^3/g) is the distribution coefficient. Note that in this case $\Psi(c)$ is not Lipschitz continuous at $c = 0$ for $0 < p < 1$.

For more details on these models, and the mathematical ramifications, see [7].

III. NUMERICAL APPROACH

Under the assumptions given above, (1) can be written as

$$\phi c_t + R(c^p)_t + \bar{u}c_x - \bar{D}c_{xx} = 0, \quad x > 0, \ t > 0, \tag{4}$$

where $R = \rho K_d/\phi$, $\bar{u} = u/\phi$, and $\bar{D} = D/\phi$ Let $\mu = c + Rc^p$, and let $\eta(\mu)$ be the inverse function, $\eta(\mu(c)) = c$ Then (4) can be written as a parabolic equation in μ:

$$\mu_t + \bar{u}\eta(\mu)_x - \bar{D}\eta(\mu)_{xx} = 0, \quad x > 0, \ t > 0. \tag{5}$$

As mentioned before, this equation poses the difficulties that the nonlinearity c^p is non-Lipschitz at $c = 0$, and the flow is generally advection-dominated, thus the solution exhibits sharp fronts. The effects of choosing $p < 1$ (as opposed to $p = 1$) are to make the fronts even sharper, and to further retard the flow of the chemical species. In fact, in the limit of zero diffusion with $p < 1$, solutions to (5) can exhibit shocks for smooth initial data.

For advection-dominated flow problems such as (5), we have studied the application of higher-order Godunov-mixed methods. This class of methods was formulated and analyzed for nonlinear advection-diffusion equations in [8]. A multidimensional extension of the method is described in [9]. We now describe the application of this algorithm to (5).

Assume we truncate our computational domain to a region $(0, \bar{x})$. At the point $\bar{x}$ we will assume the "outflow" boundary condition

$$\mu_t + \bar{u}\eta(\mu)_x = 0, \quad \text{at } x = \bar{x}. \tag{6}$$

Let $0 = x_{1/2} < x_{3/2} < \ldots < x_{J+1/2} = \bar{x}$ be a partition of $[0, \bar{x}]$ into grid blocks $B_j = [x_{j-1/2}, x_{j+1/2}]$, and let x_j be the midpoint of B_j, $h_j = x_{j+1/2} - x_{j-1/2}$, and $h_{j+1/2} = (h_j + h_{j+1})/2$. Let $\Delta t > 0$ denote a time-stepping parameter, and let $t^n = n\Delta t$. For functions $g(x,t)$, let $g_j^n = g(x_j, t^n)$.

On each grid block B_j, we approximate μ^n by a piecewise linear function $\bar{\mu}^n$, where

$$\bar{\mu}^n|_{B_j} = \bar{\mu}_j^n + (x - x_j)\delta\bar{\mu}_j^n; \tag{7}$$

c^n is approximated by a piecewise constant function C^n, where

$$C^n|_{B_j} = C_j^n. \tag{8}$$

In (7),

$$\bar{\mu}_j^n \equiv C_j^n + R(C_j^n)^p. \tag{9}$$

Let $\gamma(x,t)$ denote the diffusive flux, $\gamma(x,t) = -\bar{D}\eta(\mu)_x \equiv -\bar{D}c_x$. Applying the mixed finite element method to the diffusion terms in (5) with the lowest order Raviart-Thomas approximating spaces, and using the appropriate quadrature rule, γ is approximated by (see [8]),

$$\gamma(x_{j+1/2}, t^n) \approx \bar{\gamma}_{j+1/2}^n = -\bar{D}\frac{C_{j+1}^n - C_j^n}{h_{j+1/2}}, \tag{10}$$

for $j = 1, 2, \ldots, J - 1$. At the inflow boundary,

$$\bar{\gamma}_{1/2}^n = -2\bar{D}\frac{C_1^n - c_0^n}{h_1}. \tag{11}$$

We discuss the handling of the outflow boundary condition (6) below.

Discretizing (5) we obtain the difference equation

$$\frac{\bar{\mu}_j^{n+1} - \bar{\mu}_j^n}{\Delta t} + \bar{u}\frac{\eta(\mu_{j+1/2}^{n+1/2}) - \eta(\mu_{j-1/2}^{n+1/2})}{h_j} + \frac{\bar{\gamma}_{j+1/2}^{n+1} - \bar{\gamma}_{j-1/2}^{n+1}}{h_j} = 0, \tag{12}$$

which holds for $j = 1, \ldots, J-1$. In block B_J, we first compute the predictor $(\bar{\mu}^P)_J^{n+1}$ by

$$\frac{(\bar{\mu}^P)_J^{n+1} - \bar{\mu}_J^n}{\Delta t} + \bar{u}\frac{\eta(\mu_{J+1/2}^{n+1/2}) - \eta(\mu_{J-1/2}^{n+1/2})}{h_J} = 0. \tag{13}$$

We define $C^{n+1}(\bar{x}) = \eta((\bar{\mu}^P)_J^{n+1})$, and the diffusive flux at $\bar{x}$ is approximated by

$$\bar{\gamma}_{J+1/2}^{n+1} = -2\bar{D}\frac{C^{n+1}(\bar{x}) - C_J^{n+1}}{h_J}. \tag{14}$$

Finally, we update $\bar{\mu}_J^{n+1}$ by an equation of the form (12), with $j = J$.

The term $\eta(\bar{\mu}_{j+1/2}^{n+1/2})$ is an approximation to $\eta(\mu(x_{j+1/2}, t^{n+1/2}))$ We approximate this term by characteristic tracing from the point $(x_{j+1/2}, t^{n+1/2})$ back to time t^n, i.e., we define the characteristic $x(t)$ satisfying

$$x'(t) = \bar{u}\eta'(\bar{\mu}_j^n), \quad x(t^{n+1/2}) = x_{j+1/2}. \tag{15}$$

Assuming the CFL constraint

$$\lambda_j^n = \frac{\Delta t}{h_j}\bar{u}\eta'(\bar{\mu}_j^n) \leq 1, \tag{16}$$

then $x(t)$ crosses the $t = t^n$ axis at a point $x_{j,L} \in B_j$, where

$$x_{j,L} = x_{j+1/2} - \frac{\Delta t}{2}\bar{u}\eta'(\bar{\mu}_j^n). \tag{17}$$

The term $\eta(\bar{\mu}_{j+1/2}^{n+1/2})$ is given by

$$\eta(\bar{\mu}_{j+1/2}^{n+1/2}) \equiv \eta(\bar{\mu}^n(x_{j,L})). \tag{18}$$

Thus, the advective part of (12) is handled fully explicitly. Substituting (9), (18), (10), (11), and (14) into (12), we obtain a nonlinear system of equations in $C_j^{n+1}, j = 1, \ldots, J$. Once C_j^{n+1} is determined, $\bar{\mu}_j^{n+1}$ is updated by (9).

The last step in the calculation at time t^{n+1} is the computing of the slopes, $\delta\bar{\mu}_j^{n+1}$, $j = 1, \ldots, J$. The slope in the last interval, $\delta\bar{\mu}_J^{n+1}$ is set to zero. In the remaining intervals we set

$$\delta\bar{\mu}_j^{n+1} = \delta_{lim}\bar{\mu}_j^{n+1} \cdot sign(\bar{\mu}_{j+1}^{n+1} - \bar{\mu}_{j-1}^{n+1}), \tag{19}$$

where

$$\delta_{lim}\bar{\mu}_j^{n+1} = \begin{cases} \min(|\Delta_+\bar{\mu}_j^{n+1}|, |\Delta_-\bar{\mu}_j^{n+1}|), & \text{if } \Delta_+\bar{\mu}_j^{n+1} \cdot \Delta_-\bar{\mu}_j^{n+1} > 0, \\ 0, & \text{otherwise .} \end{cases} \tag{20}$$

Here $\Delta_+\bar{\mu}_j$ is the forward difference $(\bar{\mu}_{j+1} - \bar{\mu}_j)/h_{j+1/2}$, and $\Delta_-\bar{\mu}_j$ is the corresponding backward difference. The point of the procedure (19)-(20) is to compute a piecewise linear approximation which doesn't without introducing new extrema into the approximate solution. Thus, in blocks where the solution already has a local extrema, the slope $\delta\bar{\mu}_j$ is set to zero.

IV. NUMERICAL RESULTS

In this section, we study the effect on the solution of varying the exponent p. We choose $c_0 \equiv 1$, $\phi = .5$, $R = 1.5$, $u = 2.5\ cm/h$, and $D = .15\ cm^2/h$. The computational domain is $0 < x \leq 100\ cm$. The initial condition for all cases is plotted in Figure 1.

We first consider the case $p = .8$. To test the convergence of the scheme, we compare the approximate solutions at time $t = 25$ hours, generated using 50 and 100 grid blocks. As seen in Figure 1, these solutions are very close In Figure 2, we compare solutions for $p = .5$, .8, and 1 at $t = 25$ hours. This figure shows that increasing p results in sharper fronts and substantial retardation of the solution, as expected.

In conclusion, the Godunov-mixed method approach described here gives solutions which agree with physical intuition. In future work, we will extend the method to model more physically interesting situations in multiple space dimensions.

REFERENCES

[1] J. M. Thomas, M. D. Lee, P. B. Bedient, R. C. Borden, L. W. Canter, and C. H. Ward, *Leaking underground storage tanks: remediation with emphasis on* in situ *biorestoration*, Environmental Protection Agency, 600/2-87,008, January, 1987.

[2] United States Department of Energy, *Site-directed subsurface environmental initiative, five year summary and plan for fundamental research in subsoils and in groundwater, FY1989-FY1993*, DOE/ER 034411, Office of Energy Research, April 1988.

[3] M. F. Wheeler and C. N. Dawson, *An operator-splitting method for advection-diffusion-reaction problems*, MAFELAP Proceedings VI, J. A. Whiteman, ed., Academic Press, pp. 463-482, 1988.

[4] C. Y. Chiang, C. N. Dawson, and M. F. Wheeler, *Modeling of* in-situ *biorestoration of organic compounds in groundwater*, to appear in Transport in Porous Media.

[5] C. N. Dawson and M. F. Wheeler, *Characteristic methods for modeling nonlinear adsorption in contaminant transport*, Proceedings, 8th International Conference on Computational Methods in Water Resources, Venice Italy, 1990, Computational Mechanics Publications, Southampton, U. K., pp. 305-314.

[6] J. J. T. I. Boesten, *Behaviour of herbicides in soil. simulation and experimental assesment*, Ph. D. Thesis, Wageningen, 1986.

[7] C. J. van Duijn and P. Knabner, *Solute transport in porous media with equilibrium and non-equilibrium multiple-site adsorption: Travelling waves*, Institut für Mathematik, Universitat Augsburg, Report No. 122, 1989.

[8] C N. Dawson, *Godunov-mixed methods for advective flow problems in one space dimension*, to appear in SIAM J. Numer. Anal.

[9] C. N. Dawson, *Godunov-mixed methods for immiscible displacement*, International Journal for Numerical Methods in Fluids 11, pp. 835-847, 1990.

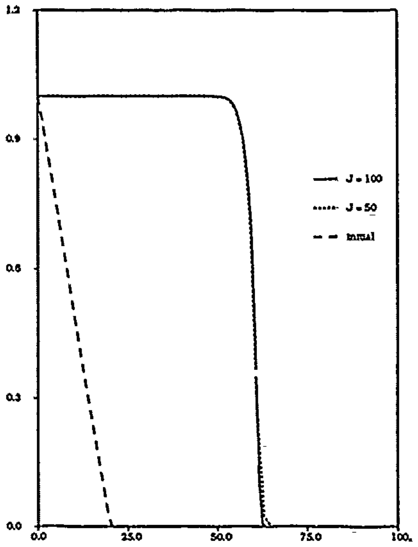


Figure 1: Test of convergence for $p = .8$.

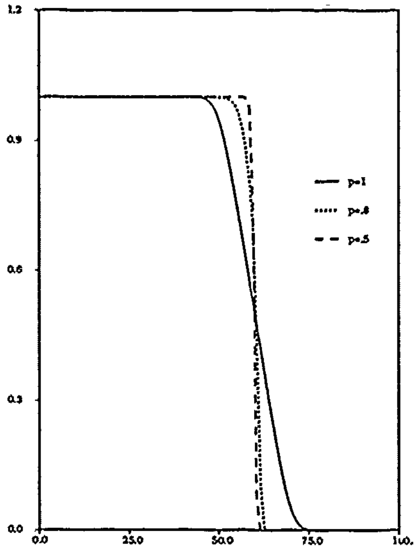


Figure 2: Comparison of $p = .5, .8$, and 1.

EFFECTIVE DISPERSION MODELS FOR VISCOUS FINGERING IN HETEROGENEOUS MEDIA

Richard E. Ewing
Departments of Mathematics, Chemical Engineering,
and Petroleum Engineering
University of Wyoming
Laramie, Wyoming 82071

Abstract–In order to scale the highly localized behavior of viscous fingering generated by heterogeneous media up to computational and field scales, we must develop techniques to obtain effective parameters for coarse-grid models which match fine-grid simulations. In [1], Russell, Young, and the author presented a coarse-grid dispersion model of heterogeneity and viscous fingering to match fine-grid simulation of miscible displacement processes. They adjusted longitudinal and transverse dispersivities in a dispersion tensor to match recovery curves for simulations of viscous fingering on fine grids. Although they were able to match production from various simulations, they pointed out that permeability averages, variances, and standard deviations alone are not able to determine dispersivities, since the specific permeability distribution in each realization can have significant impact upon the flow and hence the recovery. In [2], Espedal *et al.* consider similar dispersion models for immiscible, two-phase flow. In this paper we combine these ideas for multiphase and multicomponent flow, using dispersion models coupled with accurate treatment of first-order transport effects for both models. This coupling will be very important for fully compositional models, which possess aspects of each process. The dispersion models are presented for both multicomponent and multiphase cases. Then accurate high-resolution numerical simulators are introduced and used as our experimental tool. Numerical results illustrate the success of dispersion models for all these problems.

I. INTRODUCTION

The understanding and prediction of the behavior of the flow of multiphase or multicomponent fluids through porous media are often strongly influenced by heterogeneities or quite localized phenomena. Although considerable information can be gained about the physics of multiphase flow of chemically reacting fluids through porous media via laboratory experiments and pore-scale models, the length scale of these data is quite different from that required for field-scale understanding. The coupled fluid/fluid interactions are highly nonlinear and quite complex. The presence of heterogeneities in the medium greatly complicates this flow. We must understand the effects of heterogeneities coupled with nonlinear parameters and functions on different length scales. We use the simulators as "experimental tools" in the laboratory of supercomputer environments to simulate the process on increasingly larger length scales to develop intuition on how to model the effects of heterogeneities and viscous fingering at various levels.

II. DISPERSION MODELS

In order to ensure that the information passed from scale to scale is dependent upon the physical properties of the flow and not upon the numerics of the specific simulator, we have extensively studied the codes used and have shown them to be essentially free of numerical dispersion and grid orientation effects. The codes utilize mixed finite element methods for accurate fluid velocities in the presence of heterogeneities and modified method of characteristics techniques for accurate fluid transport without numerical dispersion.

The miscible displacement of one fluid by another in a porous medium Ω is given by

$$\phi\frac{\partial c}{\partial t} + \nabla \cdot uc - \nabla \cdot D\nabla c = q\tilde{c}, \qquad x \in \Omega,$$

where c, a fraction between 0 and 1, is the concentration of the invading fluid, ϕ is the porosity of the medium, u is the fluid velocity, q is the flow rate at the wells, $\tilde{c}$ is the resident concentration at the well and D is the dispersion tensor given by

$$D = \phi\left(d_m I + d_\ell|u|E + d_t|u|E^{\perp}\right);$$

here d_m, d_ℓ, and d_t are the molecular, longitudinal, and transverse dispersivities, respectively, $e_{ij} = u_i u_j/|u|^2$, and $E^{\perp} = I - E$. Both miscible and immiscible codes used in our simulations utilize a physical dispersion tensor with different longitudinal and transverse terms. Usually d_ℓ is approximately ten times d_t. Although this is clearly natural for miscible displacement, the local physics of multi-phase flow does not normally involve a dispersion phenomena. However, via perturbation analysis, Espedal has developed a natural dispersion tensor arising from heterogeneous flow at larger length scales. Furtado *et al.* [3] have stochastically arrived at a dispersion phenomenon with effects somewhere between transport and diffusion in origin. This corresponds to the need to match the gross permeability effects with first-order transport concepts and the finer-scale fingering with dispersion models.

III. NUMERICAL EXPERIMENTS

In the numerical experiments, we systematically vary mobility ratio, longitudinal and transverse dispersivity, and heterogeneity on fine grids. We use log normal permeability distributions, considering the effect of variance. We also simulate several different randomly generated permeabilities with the same statistical properties to see whether the gross fingering behavior and recovery are similar. Then we seek relationships between the fine grid parameters and those in the coarse grid models to use effective parameters which match "averaged" properties of many fine grid simulations. The computations for both the multicomponent and multiphase models on fine grids have been matched effectively via dispersion models.

REFERENCES

1. R.E. Ewing, T.F. Russell, and L.C. Young, An anisotropic coarse-grid dispersion model of heterogeneity and viscous fingering in five-spot miscible displacement that match experiments and fine-grid simulations, *Proceedings 10th SPE Symposium on Reservoir Simulation*, Houston, Texas (1989), 447–466, and *SPE Res. Eng.*, (to appear).

2. M.S. Espedal, P. Langlo, D. Sævareid, E. Gislefoss, and R. Hansen, Heterogeneous reservoir models, local refinements, and effective parameters, *SPE 21231, Proceedings of Eleventh SPE Symposium on Reservoir Simulation*, Anaheim, California (1991), 307–316.

3. J. Furtado, J. Glimm, W.B. Lindquist, and L.F. Pereira, Characterization of mixing length growth for flow in heterogeneous porous media, *Proceedings of Eleventh SPE Symposium on Reservoir Simulation*, Anaheim, California (1991), 317-322.

Reentry Aerothermodynamic Simulations using the Taylor-Galerkin Finite-Element Method

E. Laurien, M. Böhle, H. Holthoff and J. Wiesbaum

Institute for Fluid Mechancis
Technical University of Braunschweig
3300 Braunschweig, Bienroder Weg 3, Germany

Abstract- The hypersonic flow around space capsule-like bodies under the conditions of reentry into the earth's atmosphere is simulated. As an appropriate numerical algorithm the Taylor-Galerkin finite-element method has been selected. Results are presented for flows in thermodynamical and chemical equilibrium.

I. INTRODUCTION

The engineering need to simulate reentry aerothermodynamics requires the development of new numerical algorithms and their application to realistical configurations. The physical problem involves strong shocks, high temperatures, boundary layers, entropy layers and the interaction of these phenomena. Therefore a numerical method to simulate such flows must posess flexible spatial approximation properties, such as provided by unstructured locally refined computational grids. Furthermore it is desirable to couple the flow computation with computations of the heat flow within the heat-shield consisting of non-ablative ceramic material. We have chosen the Taylor-Galerkin finite-element method [1,2] as the basis for the development of a three-dimensional aerothermodynamical simulation code. In the present paper some computational aspects are outlined and new three-dimensional simulations under reentry conditions are presented.

II. GRID GENERATION

An axisymmetric three dimensional grid around bodies of revolution is generated using the cylindrical coordinate system x,r,φ. First a relatively coarse two-dimensional unstructured grid consisting of triangles in the x-r-plane is computed using transfinite interpolation to generate the nodes and Delaunay triangulation [3] to generate the triangles. The grid is then locally refined in regions of shocks and boundary layers by subdivision of selected triangles with 'hanging nodes' avoided. This grid is then rotated around the axis $r = 0$ by small angles φ_n forming an array of n sectors between successive planes $0 \le \varphi \le 2\pi$. Corresponding triangles of these planes form skewed prisms, each of which is subdivided into three tetrahedrons in space. By this technique large three-dimensional grids consisting of tetrahedrons can be generated, the tetrahedrons of each plane being geometrically similar. The grid is 'semistructered', i.e. unstructured in x,r and structured in φ.

III. SIMULATION ALGORITHM

As the algorithm to simulate aerothermodynamics of reentry including strong shocks, boundary layers, entropy layers, and thermodynamical and chemical relaxation the explicit two-step Taylor-Galerkin method [1,2] is applied. This algorithm is formulated in a cartesian coordinate system x,y,z using transformed node-coordinates. Elementwise constant and linear shape functions are used. Beginning with a parallel flow the Navier-Stokes and coupled chemistry equations are integrated in time. Integrals of the shape functions and their derivatives are split into two parts, the first only depending on x and r and the second on φ. In an efficient implementation of explicit finite-element methods integrals must be precomputed and stored for all permutations of element numbers, local node numbers and coordinate directions. In our method we reduce the computer space greatly by only precomputing the first part for each tetrahedron of one sector and the second for each sector. Using the axisymmetric grid nonaxisymmetric flow can be computed, e.g. with a small angle of attack.

In order to improve shock-capturing properties the algorithm of flux-corrected transport [4,5] has been implemented. A model of chemically reacting air [6] at high temperatures was so far applied to thermal and chemical equilibrium and is currently being implemented for nonequilibrium flow. In the equilibrium case the system of chemical reactions as well as the vibrational excitation of the molecules can be decoupled and solved a priory. This precomputed solution is used during the simulation as a Chebychev approximation.

IV. RESULTS

The algorithm is tested using the two-dimensional example of a circular cylinder at an inflow Mach number of 20 for frictionless flow. The computational grid and a perfect gas solution is shown in fig. 1. A three-dimensional Euler simulation of a sphere (220 000 elements, 40 sectors) is shown in figure 2.

V. CONCLUSIONS

Finite-element simulations of three-dimensional flow around capsule-like configurations can be conducted on axisymmetric semistructured grids. First results for thermochemical nonequilibrium and viscous flow will soon be available. However, physical and numerical models must be validated by comparision with experimental measurements during actual reentries.

VI. REFERENCES

[1] R. Löhner, K. Morgan, and O.C. Zienkiewicz: An Adaptive Finite Element Procedure for Compressible High Speed Flows, Comp. Meth. Appl. Mech. Eng. 51, 441 - 465 (1985)

[2] E. Laurien, M. Böhle, H. Holthoff, and M. Odendahl: Stability and Convergence of the Taylor-Galerkin Finite-Element Method for the Navier-Stokes Equations, ZAMM 71, T411 - T413 (1991)

[3] S.W. Sloan and G.T. Houlsby: An Implementation of Watson's Algorithm for Computing 2-Dimensional Delaunay Triangulations, Adv. Eng. Software 6, 192 - 196 (1984)

[4] R. Löhner, K. Morgan, J. Peraire, and M. Vahdati: Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations, Int. J. Num. Meth. Fluids 7, 1093 - 1109 (1987)

[5] E. Laurien, M. Böhle, and H. Holthoff: Numerical Approximation of Hypersonic Shocks in a Finite Element Method using Flux-Corrected Transport, GAMM Annual Meeting, April 1-4, 1991, Cracow, Poland to appear in ZAMM

[6] C. Park. On Convergence of Computation of Chemically Reacting Flows, AIAA-85-0247 (1985)

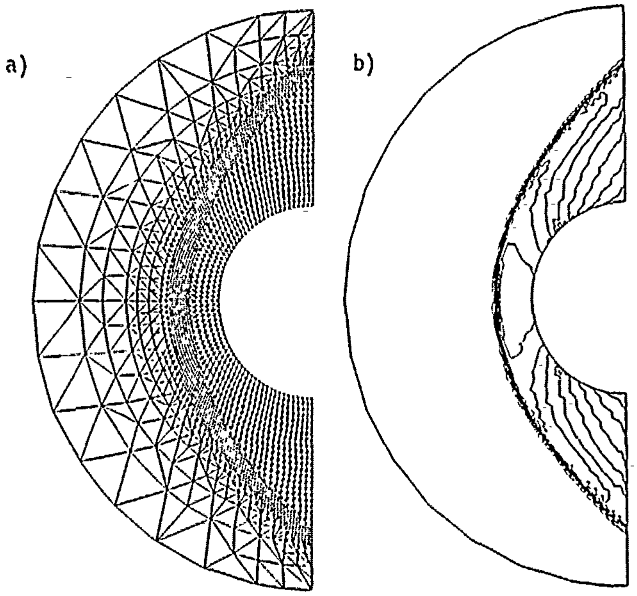


Fig. 1. Two-dimensional simulation at an inflow Mach number of 20,; a) computational grid, b) isolines of the density

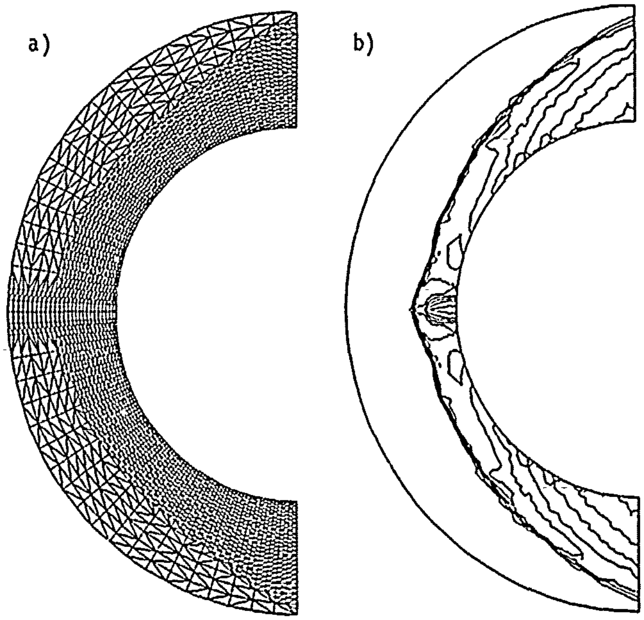


Fig. 2: Three-dimensional Euler simulation of a sphere in a Mach 6 flow, a) c.oss section of the compuational grid in the x-r-plane, b) isolines of the density

The large-scale structures of three-dimensionally evolving jets

ECKART MEIBURG
Department of Aerospace Engineering
University of Southern California
University Park
Los Angeles, CA 90089-1191 USA

and

JAMES E. MARTIN
Center for Fluid Mechanics
Division of Applied Mathematics
Brown University
Providence, RI 02912 USA

Abstract - We continue our investigation of the three-dimensional evolution of nominally axisymmetric transitional jets subject to axisymmetric, helical or azimuthal perturbations and combinations thereof. Our approach is a computational one, employing inviscid vortex filament techniques to gain insight into the mechanisms leading to the concentration, reorientation, and stretching of vorticity. Our earlier studies had demonstrated the emergence of vortex rings connected by couterrotating pairs of streamwise braid vortices for the case of superimposed axisymmetric and azimuthal perturbations. Furthermore, for the case of a helical perturbation combined with an azimuthal one, we observed the emergence of concentrated streamwise braid vortices all of the same sign. In the present investigation, we study the interaction of two helical perturbations of opposite sign. While they cancel each other at some azimuthal locations, they become amplified at others, thus leading to a complex three-dimensional flow pattern exhibiting regions of strong azimuthal vortices connected by concentrated streamwise vorticity. In addition, the interaction between the opposite-sign helices results in strong azimuthal velocities.

1. Introduction

The present investigation represents a continuation of our earlier studies of the evolution of transitional jets under three-dimensional perturbations (Meiburg, Lasheras and Martin 1989, Meiburg and Martin 1990, Martin and Meiburg 1991). The types of perturbations we have been considering are of wave-like character in the streamwise, helical and azimuthal directions, as past stability analyses had demonstrated their relevance with respect to axisymmetric jets and vortex rings (Batchelor and Gill 1962, Widnall, Bliss and Tsai 1974, and Cohen and Wygnanski 1987, to mention just a few). Recent experimental investigations by Tso and Hussain (1989) as well as Mungal and Hollingsworth (1989) show convincingly that even fully turbulent jets are dominated by ringlike and helical structures whose dynamics become largely independent of the Reynolds number when this dimensionless parameter is large. Furthermore, Corke and Kusek (1990) experimentally demonstrate the possibility of resonance in axisymmetric jets with helical mode pairs. Edwards, Marx and Ashurst (1991) observe large-scale structures of a helical nature in swirling jets as well Consequently, our series of studies aims at achieving a more complete understanding of the nonlinear growth and dynamics of these structures, with the ultimate goal of successful manipulation and control of jets. Our numerical investigation of axially forced jets emerging from a corrugated nozzle (Martin and Meiburg 1990) showed the formation of vortex rings that set up a strain field with a free stagnation point in the braid region. Small perturbations in the braid vorticity due to the corrugation are subsequently amplified, whereupon pairs of concentrated streamwise counterrotating vortices form in between the vortex rings. This scenario is in accordance with the mechanism suggested by Lin and Corcos (1984) as well as by Neu (1984) for the plane mixing layer. If, on the other hand, the axisymmetric jet is perturbed by a helical wave, a layer of streamwise braid vorticity forms that has the same sign everywhere (Meiburg and Martin 1990). If the helical symmetry is broken by introducing a periodic perturbation in the azimuthal direction, streamwise braid vortices emerge that become amplified in the strain field of the helix. Ir the present paper, we will study the inviscid evolution and interaction of a helical mode pair of azimuthal wavenumbers +/-1. The numerical technique will briefly be described in section 2. In section 3, we will discuss the emerging flowfield with particular emphasis on the large-scale vortical structure.

2. Numerical Technique

The non-divergent nature of the velocity field in incompressible flows, along with the definition of vorticity, allows for a complete description of the kinematics of the flow in the form of the Biot-Savart law. Using the theorems of Kelvin and Helmholtz for inviscid dynamics and following the general concepts reviewed by Leonard (1985), vortex filaments are used for the representation of the vorticity field. Each filament is represented by a number of node points along its centerline, through which a cubic spline is fitted to give it a smooth shape. The three-dimensional Biot-Savart integral is reduced to a line integral by assuming an invariant algebraic vorticity distribution around the filament centerline. For the numerical simulation, we limit ourselves to the temporally growing problem, i.e., our flow is periodic in the streamwise direction. We take the velocity difference between the centerline and infinity as our characteristic velocity. The thickness of the axisymmetric shear layer, defined as the velocity jump divided by the maximum slope of the velocity, serves as the characteristic length scale, which results in the filament core radius of 0.5. In these units, the radius of the jet considered here is 5. Hence, the important ratio of jet radius to momentum thickness of the jet shear layer is 22.6. The Biot-Savart integration is carried out with second order accuracy both in space and in time by employing the predictor-corrector time-stepping scheme and the trapezoidal rule for spatial integration, respectively. A more detailed discussion of the numerical method can be found in Ashurst and Meiburg (1988).

3. Results and Discussion

The subject of our investigation is a nominally axisymmetric jet perturbed by two helical waves of azimuthal wavenumbers +1 and -1, respectively. The axisymmetric shear layer is represented by vortex filaments that initially have the form of vortex rings. Consequently, there is no overall swirl in the jet. Numerically, each of the two helical perturbation waves is introduced by slightly displacing the vortex filament centerlines in the streamwise direction. Figure 1 shows the flow field at time t=1.72. The two side views and the streamwise view clearly demonstrate that the two perturbation waves cancel each other near y=0, whereas they amplify each other around z=0, thereby forming regions in which the vorticity becomes slightly more concentrated. Thus, a Kelvin-Helmholtz-type instability of the axisymmetric shear layer is triggered near z=0, which leads to a roll-up of the vorticity layer and to the formation of segments of concentrated vortex rings on opposite sides of the jet. However, these segments are out of phase with each other, so that a contour plot of the azimuthal vorticity component in the plane z=0 would show a staggered pattern, resembling a Karman vortex street. This phase shift between the emerging vortex ring segments on opposite sides of the jet leads to the interesting situation in which some regions of a vortex filament are convected towards the jet axis, whereas others are displaced away from the axis. In this fashion, the vortex filament, in between the emerging vortex ring sections,

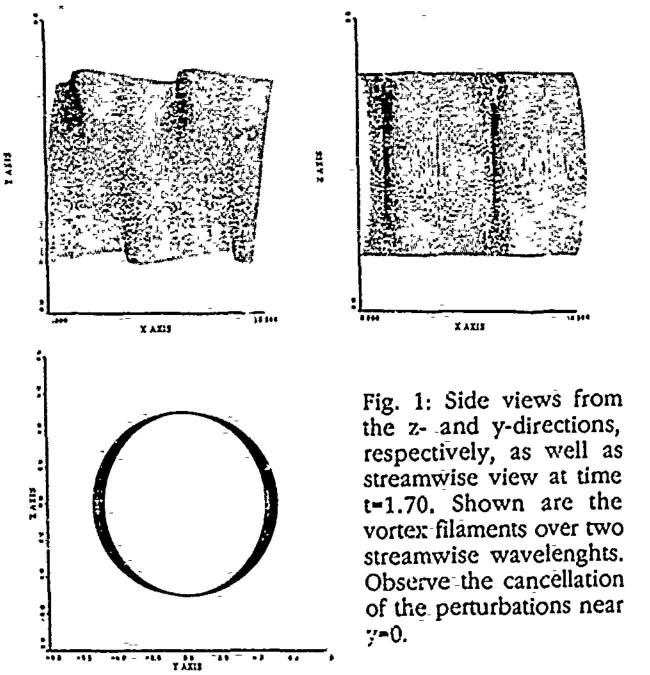


Fig. 1: Side views from the z- and y-directions, respectively, as well as streamwise view at time t=1.70. Shown are the vortex filaments over two streamwise wavelenghts. Observe the cancellation of the perturbations near y=0.

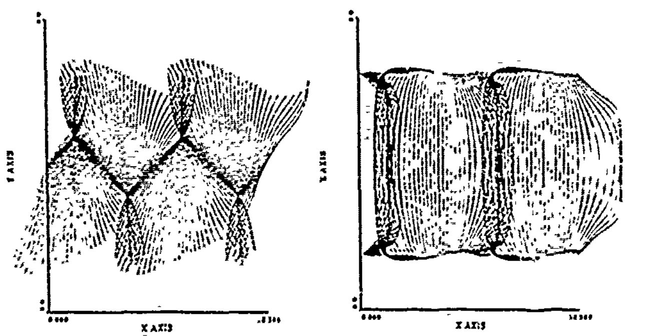


Fig. 2: z- and y-views at time t=5.70. Note the formation of out-of-phase vortex ring segments.

develops a streamwise vorticity component which becomes larger as the amplitude of the Kelvin-Helmholtz instability grows. This tendency has become much more pronounced by time t=5.70 (figure 2). We observe the formation of concentrated streamwise vortical structures connecting the out-of-phase vortex ring segments. This situation appears similar to the one in mixing layers with a defect caused by a phase jump at a given spanwise location. By time t=8.05 (figure 3), these concentrated streamwise structures have grown in strength, as they encompass an increasing number of vortex filaments. We furthermore see that they spiral around each other where they come together to form a vortex ring segment. This behavior is also clearly visible in the streamwise view, and it becomes considerably more prominent by time t=11.95 (figure 4). It creates a strong axial velocity component along the vortex ring segments. This situation locally resembles observations of helical vortex breakdown. Interesting questions concern the dependence of the flow pattern on the ratio of jet radius to jet shear layer thickness, and its stability under additional axisymmetric or azimuthal perturbations. A more detailed investigation along these lines is currently under way.

Acknowledgements

This work is supported by the National Science Foundation under grant #CTS-9058065 (to E.M.) and by DARPA under URI contract #N00014-86-K0754 to Brown University. The San Diego Supercomputer Center is providing computer time.

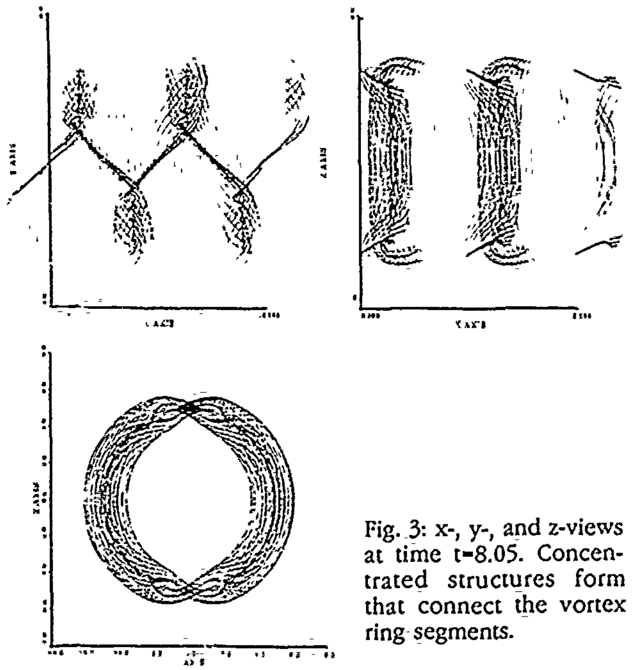


Fig. 3: x-, y-, and z-views at time t=8.05. Concentrated structures form that connect the vortex ring segments.

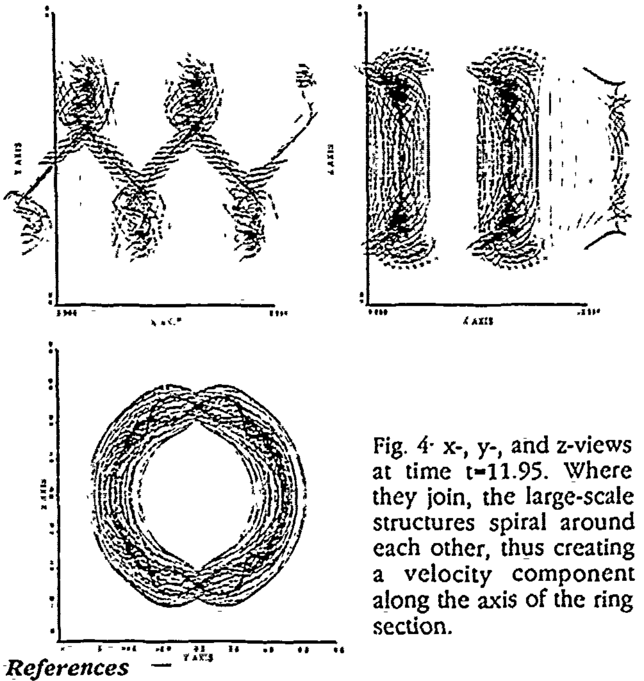


Fig. 4· x-, y-, and z-views at time t=11.95. Where they join, the large-scale structures spiral around each other, thus creating a velocity component along the axis of the ring section.

References

ASHURST, W.T. and MEIBURG, E. 1988 J. Fluid Mech. **189**, 87.
BATCHELOR, G.K. and GILL, A.E. 1962 J. Fluid Mech. **14**, 529.
COHEN, J. and WYGNANSKI, I. 1987 J. Fluid Mech. **176**, 191.
CORKE, T.C. and KUSEK, S.M. 1990 Bull. Am. Phys. Soc. **35**, 2328.
EDWARDS, C.F., MARX, K.D., ASHURST, W.T. 1991 Preprint.
LEONARD, A. 1985 Ann. Rev. Fluid Mech. **17**, 523.
LIN, S.J. and CORCOS, G.M. 1984 J. Fluid Mech. **141**, 139.
MARTIN, J.E. and MEIBURG, E. 1991 To appear. J. Fluid Mech.
MEIBURG, E. and MARTIN, J.E. 1990 To appear. Advances in Turbulence 3, Springer.
MEIBURG, E. , LASHERAS, J.C., and MARTIN, J.E. 1989 To appear: Turbulent Shear Flows VII, Springer.
MUNGAL, M.G. and HOLLINGSWORTH, D.K. 1989 Phys. Fluids A **1**, 1615.
NEU, J.C. 1984 J. Fluid Mech. **143**, 253.
TSO, J. and HUSSAIN, F. 1989 J. Fluid Mech. **203**, 425.
WIDNALL, S.E., BLISS, D.B., and TSAI, C.-Y. 1974 J. Fluid Mech. **66**, 35.

ASPECTS OF THE NUMERICAL COMPUTATION OF HYPERSONIC REACTIVE FLOW

Jean-Antoine Désidéri

INRIA Centre de Sophia Antipolis
2004 Route des Lucioles, 06560 Valbonne - France

General considerations on hypersonic reactive flow and the challenge of computation

The hypersonic flight of a space vehicle when it reenters the atmosphere (high Mach number, up to 30, high angle of attack, up to 30°) corresponds to a critical phase from the viewpoint of aerodynamical control and thermal loads control, and gives rise to the development of advanced numerical specialized simulation tools.

At high altitude, the low-density atmosphere (p_∞=2.52 Pa, ρ_∞=4.28 10^{-5} kg/m^3 at the altitude of 75 km) undergoes a strong compression across the main detached shock of the external flow. Immediately behind this shock, assuming a freestream Mach number of the order of 30, the temperature reaches several tens of thousands degrees (K) and air dissociates and is in strong chemical but also vibrational non-equilibrium. Along a particle path, the various non-equilibrium modes relax to equilibrium at different time scales: vibrational equilibrium is first reached, then chemical equilibrium. The dissociation reactions being endothermic, they absorb an important fraction of the energy; in this process, the temperature decreases rapidly. To realize the quantitative importance of this effect, it is instructive to consider the case of a steady inviscid (external) flow for which, when chemistry is negligible (i.e. at lower Mach numbers, $M_\infty < 10$), the temperature at the stagnation point T_S can be related to the freestream temperature T_∞ by the relation

$$T_S \;/\; T_\infty \;=\; 1 + \frac{\gamma - 1}{2}\, M_\infty^2$$

which expresses the conservation of total enthalpy per unit mass along a streamline. Trying now this formula with $M_\infty = 25$, $\gamma = C_p/C_v = 7/5$ (diatomic gas) and $T_\infty = 205$ K (standard atmosphere at an altitude of 75 km) yields $T_S = 25\ 830$ K! In reality, in this inviscid case, air is completely dissociated in the stagnation-point region, and the temperature is (only) near 6000 K, that is, 4 times smaller.

Another important effect of chemistry on a typical blunt body flow is that it modifies noticeably the shock location; the stand-off distance may be reduced in some cases of 40 % or more; more generally speaking, the entire shock layer is thinner. This effect is also of great concern to the designer, since the intersection of main shock with parts of the structure should be avoided (overheating, destruction).

For all of these reasons, and since laboratory experimentation of this extreme regime is very difficult (and costly) and often impossible today for certain configurations, it is important to develop efficient and validated numerical tools for the prediction of hypersonic reactive flow.

From a numerical point of view, the challenge of this task resides in the necessity of accounting for more complex physical models (more-or-less complete chemistry models, some of which being currently developed on the basis of more recent data; inclusion of wall effects, etc.) and accounting for more complex aerodynamics: e.g. the presence of stronger shocks in particular, makes the question of robustness more critical and partly for this reason, many authors employ upwind schemes for which the artificial viscosity is inherent to the approximation. Then, the extension of known schemes to the reactive-flow case is usually not immediate, since the diagonalization of the convective terms (Euler terms) in the governing equations depends on the form of the state equation and is different when the fluid is made of several species. Several extensions of the van Leer flux-vector splitting and of the Roe and Osher flux-difference splittings in particular can now be found in the literature.

From the designer point of view, the flows of greatest interest are those in the "near-equilibrium regime". For reasons that will be discussed in the lecture, the numerical discretization of classical type of the equations governing non-equilibrium flow is very stiff in this case, and overcoming this difficulty may reveal the greatest challenge to the computational scientist.

Finally, since shock layers in the hypersonic regime are thinner than in the better-known supersonic regime due to both effects of larger Mach number and chemical dissociation, and since the physical phenomena are more complex (dissociation, vibration, etc) and more intensive (stronger shocks), it is evident that the control of the quality of the discrete approximation is more critical and is also more strongly dependent on the quality of the employed mesh. Hence, issues such as mesh generation, and more generally, meshsize control become more essential.

Brief description of the lecture

The lecture will emphasize some of the most important aspects of the numerical computation of hypersonic external flow by upwind schemes applicable to arbitrary unstructured meshes.

After the presentation of some general considerations on hypersonic reactive flow, basic equilibrium and non equilibrium models will be described. The essential effect of the Damköler number on a typical blunt-body non-equilibrium flow will be briefly discussed.

Hybrid Finite Volume/Finite-Element upwind schemes will be described for both models. Weakly-coupled (equivalent-approach") and strongly-coupled formulations will be compared. Many blunt-body flow computations will be shown to evaluate or verify numerically the effect of various parameters, such as the freestream Mach number or the size of the obstacle.

Implicit timestepping will be presented, and its efficiency demonstrated. The construction of quasi-second-order schemes incorporating monotonicity devices will be introduced with some emphasis on the appropriate choice of the set of physical variables on which limitation should be applied. In particular, the drastic effect on the wall chemical composition of the order of accuracy of the approximation scheme will be shown.

The difficulty to correctly simulate the stagnation-point region in a non-equilibrium flow will be discussed and illustrated.

Many examples of computations will deal with inviscid flow. However, a preliminary assessment of the effect of the transport model on a Navier-Stokes reactive flow computation will be included.

Finally, some considerations on mesh generation will be made. An example will be given in which several meshes have been constructed in the course of the solution convergence. Initially, one constructs a smooth (structured) mesh employing a "hyperbolic grid generator" in which the distribution of cell areas is controlled to target a prescribed external domain boundary. Then, one or more mesh enrichments are made by element division to adapt the (now unstructured) mesh to the solution. The same solver (adapted to unstructured data base) is employed throughout.

References

[1] J.A. DESIDERI, N. GLINSKY and E. HETTENA, "Hypersonic Reactive Flow Computations", *Computers & Fluids* Vol. 18, No. 2, pp.151-182, 1990.

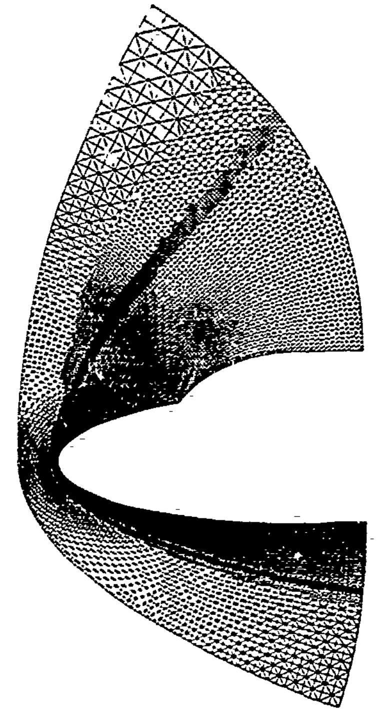
[2] N. GLINSKY, L. FEZOUI, M.C. CICCOLI, J.-A. DESIDERI, "Non equilibrium hypersonic flow computations by implicit second-order upwind finite-elements", Proc. of the *8th GAMM Conference on Numerical Methods in Fluid Mechanics, Delft, The Netherlands, September 2 -29 1989.* Notes on Numerical Fluid Mechanics Vol. 29, pp. 159-168 (Vieweg, Braunschweig, 1990).

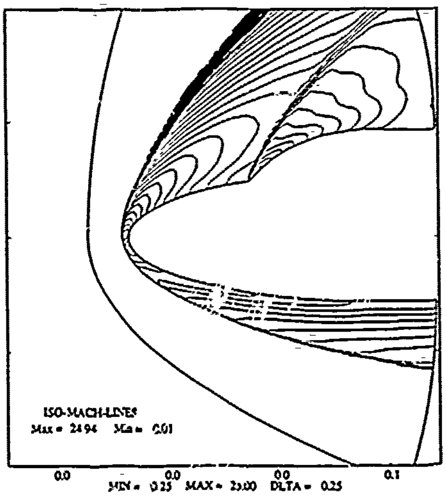
[3] N. BOTTA, M.C. CICCOLI, J.A. DESIDERI, L. FEZOUI, N. GLINSKY, E. HETTENA, C. OLIVIER, "Reactive Flow Computations by Upwind Finite Elements", and
J.A. DESIDERI, "Some Comments on the Numerical Computations of Reacting Flows over the Double-Ellipse (Double Ellipsoid)", Proc. of the *Workshop on Hypersonic Flows for Reentry Problems, Part I, January 22-25, 1990, Antibes, France,* to appear in Springer-Verlag.

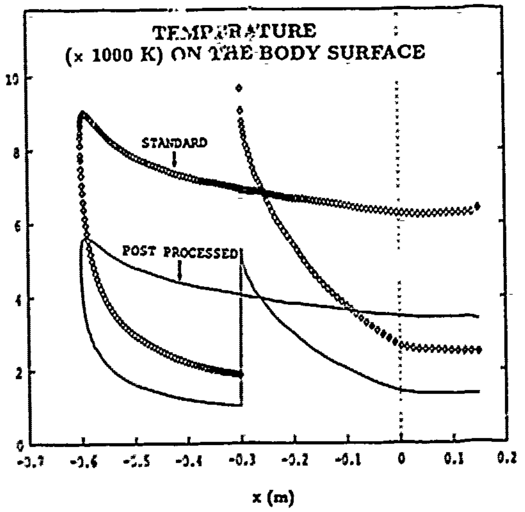
[4] M.V. SALVETTI, M.C. CICCOLI, J.A. DESIDERI, "Non-Equilibrium Inviscid and Viscous Flows over the Double-Ellipse by Adaptive Upwind Finite Elements", Presented at the *Workshop on Hypersonic Flows for Reentry Problems, Part II, April 15-19, 1991, Antibes, France.*

Illustrative Example:

Inviscid Flow over a Double ellipse (from [4])
(M_∞=25, $\alpha = 30°$, larger semi-axis = 60 cm.)







METHOD OF GRIDS ADAPTIVE TO SOLUTION FOR PROBLEMS WITH BOUNDARY LAYERS

L.M. DEGTYAREV, T.S. IVANOVA
Keldysh Institute of Applied Mathematics USSR Academy of Sciences,
Miusskaja Pl.4, Moscow 125047, USSR

I. INTRODUCTION

In mathematical physics problems an error of the finite-difference method $\|z\| = \|u-y\|$ (u is the solution of an original differential problem, y is the solution of a finite-difference problem) is determined by the number of grid points N so that as $N \to \infty$ the equality

$$\| z \| = C \left[\frac{1}{N}\right]^{m}$$

takes place asymptotically. The value of C on real "crude" grids may be decreased by redistributing the grid points. Without a priori information on the solution structure the optimal grid can not be constructed. In a general case such information can be obtained when solving the problem numerically, and then the grid can be property corrected. It is natural to call such a technique the method of grids adaptive to the solution. In this paper this technique is based on minimizing the truncation error [1]. In the method the requirement of truncation error minimization leads to the equations for the grid points coordinates. Stationary diffusion-convection problems including those with a small parameter at leading derivative are considered. In such problems the solution has high gradients in the region of boundary layers. Using the difference schemes of the second truncation order in the convective terms is not expedient here due to ocsillations of the difference solutions. The first order upwind schemes yield rather low accuracy. In the paper this contradiction is proposed to solve by increasing the upwind scheme accuracy on adaptive to solution grid. One [2] and two [3] dimensional boundary value problems are considered.

II. 1-DIMENSIONAL BOUNDARY VALUE PROBLEMS

Consider the boundary value problem

$$(\varepsilon u')' + (p u)' - q u = - f ,$$
$$u (0) = 0 , \quad u (1) = 1 , \qquad (1)$$
$$\varepsilon (x) > \varepsilon_0 , \; q (x) > q_0$$

The problem (1) is approximated by

$$(L^h y)_i = \frac{1}{\hbar_i}\left[(\varepsilon\, y_x)_{i+1/2} - (\varepsilon\, y_x)_{i-1/2}\right] +$$
$$+ \frac{1}{\hbar_i}\left[w_{i+1/2} - w_{i-1/2}\right] - q_i\, y_i = - f_i \qquad (2)$$
$$i = 1,\dots,N-1 , \; y_0 = 0 , \; y_N = 1 ,$$
$$h_{i+1/2} = x_{i+1} - x_i , \quad \hbar_i = 0.5 (h_{i+1/2} + h_{i-1/2})$$
$$y_{xi+1/2} = \frac{(y_{i+1} - y_i)}{h_{i+1/2}} , w_{i+1/2} = p^{+}_{i+1/2} y_{i+1} + p^{-}_{i+1/2} y_i$$
$$p^{\pm}_{i+1/2} = 0.5 \left[p\,(x_{i+1/2}) \pm | p\,(x_{i+1/2})|\right] .$$

We emphasize, that convective term $(p u)'$ in (2) is approximated by the first order upwind difference. It ensures the maximum principle for (2) and removes the oscillations in difference solutions. The difference solution error $z_i = y_i - u (x_i)$ satisfies the difference equation

$$(L^h z)_i = - \psi_i$$

with truncation error

$$\psi_i = \frac{1}{\hbar_i}\left[K_{i+1/2} h_{i+1/2} - K_{i-1/2} h_{i-1/2}\right] + O(\hbar_i^{\,4}) \quad (3)$$

Introduce the point grid coordinate $x (a)$ such that $x_i = x (a_i)$, $a_i = i\, h_a$ and require that

$$\psi_i = O (\hbar_i^{\,4}) \qquad (4)$$

at each point i due to a proper choice of x_i . The condition (4) may be rewritten in the form

$$(l^h x)_i = \lambda_{i+1/2}(x_{i+1} - x_i) - \lambda_{i-1/2}(x_i - x_{i-1}) = 0 \quad (5)$$

$$\lambda_{i+1/2} = |K_{i+1/2}| + \alpha_{i+1/2} , \; x_0 = 0 , \; x_N = 1 . (6)$$

The given grid function $\alpha_{i+1/2} > 0$ (in the simplest case $\alpha_{i+1/2} = \alpha$) makes it possible to redistribute the points between the high and low solution change regions. To reserve an order of truncation error the regularization function $\alpha_{i+1/2}$ in (6) should be compensated in the right hand side of the difference scheme (2) :

$$(L^h y)_i = -\left[f + \frac{1}{\hbar_i}\left[(\alpha\, h)_{i+1/2} - (\alpha\, h)_{i-1/2}\right]\right] \quad (7)$$

Different simplified modifications of grid monitor (6) are possible as well.Going over to the second order simplifies (6) essentially

[2] so that (6) takes the form

$$\lambda_{i+1/2} = \frac{1}{2}\ |p_{i+1/2}\ \frac{y_{i+1}-y_i}{x_{i+1}-x_i}\ | + \alpha \qquad (8)$$

III. TWO DIMENSIONAL BOUNDARY VALUE PROBLEMS ON TRIANGULAR GRIDS

Consider the 2D problem

$$\nabla\cdot(\varepsilon\ \nabla u) + \nabla\cdot(\vec{p}\ u) = 0,\ x = (x_1,x_2) \in \Omega\ , \qquad (9)$$

$$u + \mu\,\frac{\partial\ u}{\partial\ n} = \nu\ ,\ x \in \partial\ \Omega\ ,$$

where $\vec{p} = (p_1,p_2)$ is given vector, which satisfies

$$\nabla\cdot\vec{p} = 0\ . \qquad (10)$$

The problem (9),(10) may be considered as a model for Navier–Stokes equations. Briefly outline the difference scheme and the derivation of the grid equation for problem (9). They generalize 1D equations (7) and (8). The standart finite element approximation of (9) on an arbitrary triangular grid may be written for point i (see fig.1) as

$$\sum_k B_k(\ y_k - y_i) + \sum_k A_k\ \frac{y_k + y_i}{2} = 0 \qquad (11)$$

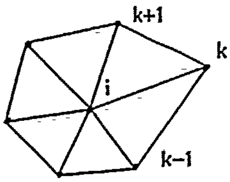


fig. 1
Element of triangular grid for scheme at point i .

In (11) the coefficients A_k , B_k depend on coefficients $\varepsilon(x)$, $\vec{p}(x)$ in the original equation and on grid geometry. On the triangular grid with acute angles only coefficients B_k always are positive. The coefficients A_k may have any sign, therefore the scheme (11) does not satisfy the maximum principle. Write (11) in the form

$$\sum_k B_k(y_k - y_i) + \sum_k (\ A_k^+\ y_k + A_k^-\ y_i) -$$
$$- \frac{1}{2}\sum_k |A_k|(y_k - y_i) = 0\ ,\ A_k^{\pm} = \frac{A_k + |A_k|}{2} \qquad (12)$$

Separate equation (12) into two equations

$$(L^h y)_i = \sum_k B_k(y_k - y_i) + \sum_k A_k^+\ y_k + A_k^-\ y_i = 0 \qquad (13)$$

$$\frac{1}{2}\sum_k |A_k|\ (y_k - y_i) = 0\ . \qquad (14)$$

The difference equation (13) is a generalization of 1D upwind difference scheme (2) onto triangular grids. It satisfies the maximum principle. From (14) we may obtain equations for grid points coordinates like (8). In fig.2 the example of adaptive grid for (10) is shown. There are two boundary layers here. The first one is located near the up boundary of width ~ $\sqrt{\varepsilon}$, the second – near the right boundary of width ~ ε .

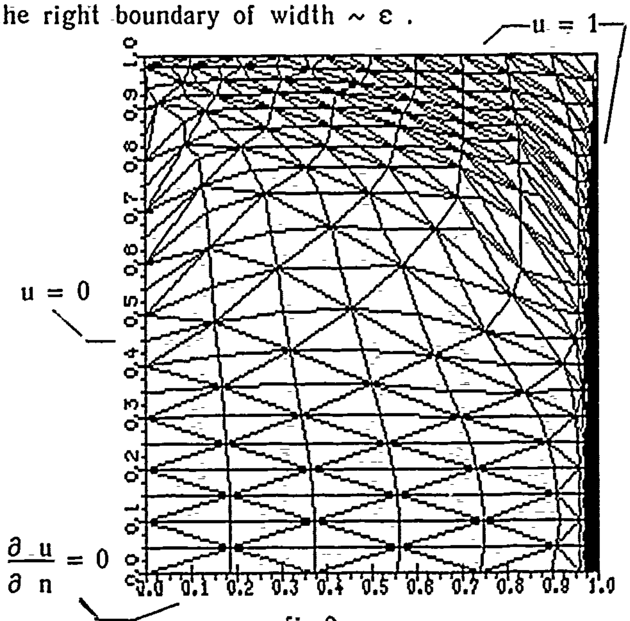


fig.2
Adaptive grid for equation $\varepsilon\Delta u - \frac{\partial u}{\partial x} = 0$, ε=0.01. Boundary conditions are shown in fig.2.

IV. GENERALIZATIONS

Note some generalizations of the method. First, we used local optimization of truncation error. Meanwhile it is possible to use optimization of some integral norm of truncation error and to construct the grid equations. Second, in section III we considered the 2D diffusion–convection equation. This approach admits a direct generalization onto such 3D equations. Finally the proposed approach may be extended to time dependent problems. The algorithm and resuls for 1D evolution problems are given in [4].

References

1. Degtyarev L.M., Drozdov V.V., Ivanova T.S. Differentsial'nye Uravneniay, XXIII, N 1, p.1160 (in russian).

2. Degtyarev L.M., Ivanova T.S. Preprint Keldysh Inst. Appl. Mathem. N 145, 1990.

3. Degtyarev L.M., Ivanova T.S. Preprint Keldysh Inst. Appl. Mathem. 1991 (to be published).

4. Degtyarev L.M., Ivanova T.S. Preprint Keldysh Inst. Appl. Mathem. 1991 (to be published).

Invariant Manifold Theorems for the Navier Stokes Equations

S. S. Sritharan
Department of Aerospace Engineering
University of Southern California
Los Angeles, California 90089-1191

Abstract

In this paper we review the results on existence, uniqueness and regularity of invariant manifolds for the Navier-Stokes equations. Some aspects of global attractors are also discussed.

1 Introduction

We will present certain results on the structure of attractors for the Navier Stokes equations. Ladyzhenskaya [11] proved that the time dependent viscous flow in two dimensional bounded domains can be characterized by a compact global attractor defined in the following way. Let $W(t,0;\cdot)$ be the solution operator which relates the velocity field at any given time to the data. For the case of fluid flow in two dimensional bounded domains, the operator $W(t,0;\cdot)$ is defined in a Hilbert space H for all positive times as a compact operator and satisfies the semigroup property:

$$W(t,t_1;W(t_1,0;\cdot)) = W(t,0;\cdot), \quad 0 \le t_1 \le t.$$

We then note that the Navier-Stokes equations define a dissipative dynamical system. This means there exists an absorbing set in the solution space inside which all orbits enter after a certain time. Existence of absorbing set was first noted by E. Hopf in 1941 [9] using energy estimates. Let $B_R(H)$ be the absorbing ball in the Hilbert space H. Then the global attractor Λ associated with the nonlinear semigroup $W(\cdot)$ is defined as the Ω-limit set of $B_R(H)$:

$$\Lambda = \bigcap_{t\geq 0}\overline{\bigcup_{s\geq t} W(s,0;B_R(H))}.$$

The set Λ is compact, connected, attracts all bounded sets of H and is invariant to $W(t,0,\ )$ for positive as well as negative times. This attractor contains in particular the steady, periodic, quasi-periodic and almost periodic solutions. The Hausdorff dimension d_H of Λ has been shown to be finite [1, 12, 14, 21, 22, 3]. Such results for three dimensional flow problems would be of great interest for the understanding of the dynamics of turbulence. Another open problem is the task of proving the existence of global finite dimensional invariant manifolds containing the attractor. The possibility of existence of global invariant manifolds containing Λ is suggested by the theorem of Mane [17] which essentially implies that Λ can be parametrized by N_Λ number of parameters with $N_\Lambda \geq 2d_H + 1$. Coordinates of such manifolds would provide us with an effective means of computing the properties of the attractor (which may be of fractional dimension.). In this report we will present certain results in this subject. First result [25] concerns with the hyperbolicity of periodic solutions of the Navier Stokes equations in bounded domains. Analyticity of the invariant manifolds is the central result. For earlier studies on hyperbolicity see [13],[8]. The second result is the upper semicontinuity of the global attractor with respect to regularizations of the Navier Stokes equations by the addition of artificial viscosity terms. We will also present a smoothness result for global invariant (inertial) manifolds for the regularized system. These global (or inertial) manifolds can also be regarded as approximate global manifolds for the original system.

In this paper we will only consider viscous flow in bounded domains. For a discussion on various issues of unbounded domains see [25].

2 Governing equations and functional framework

Let $\Omega \subset R^n, n = 2$ or 3 be a smooth bounded domain. We will consider the problem of finding $(u,p) : \Omega \times (0,\infty) \to R^n \times R$ such that,

$$u_t + (u\cdot\nabla)u = -\nabla p + \nu\Delta u \qquad \text{in } \Omega \times (0,\infty),$$

$$\nabla \cdot u = 0 \qquad \text{in } \Omega \times (0,\infty),$$

$$u(x,t) = u_b(x,t) \text{ for } (x,t) \in \partial\Omega \times [0,\infty) \text{ with } \int_{\partial\Omega} u_b \cdot d\Sigma = 0$$

$$\text{and} \quad u(x,0) = u_0, \qquad \text{for } x \in \Omega.$$

Here if Ω is multiply connected then we require that the flux through *each* component of the boundary be zero.

Let $(U(x,t), P(x,t))$ be a basic solution field (which is an orbit on the attractor) that satisfies the boundary conditions. It is known that [11] the orbits on the attractor Λ are regular (that is U, P are infinitely smooth) and defined for positive as well as negative times. We are interested in studying the solution orbits nearby this given orbit in an appropriate function space. Let us introduce the change of variables $u = U + v$ and $p = P + q$ so that (v,q) satisfy,

$$v_t+(U\cdot\nabla)v+(v\cdot\nabla)U+(v\cdot\nabla)v = -\nabla q+\nu\Delta v \qquad \text{in } \Omega\times(0,\infty),$$

$$\nabla \cdot v = 0 \qquad \text{in } \Omega \times (0,\infty), \tag{1}$$

$$v(x,t) = 0 \qquad \text{for } (x,t) \in \partial\Omega \times [0,\infty),$$

$$\text{and} \qquad v(x,0) = v_0, \qquad x \in \Omega.$$

2.1 Functional frame work

We will use the vector spaces,

$$j(\Omega) = \{u . \Omega \to R^n; u \in C_0^\infty(\Omega), \text{div} u = 0\},$$

$$H = \{u . \Omega \to R^n, u \in L^2(\Omega), \text{div} u = 0, u(x) \cdot n = 0, x \in \partial\Omega\}.$$

$$V = \{u : \Omega \to R^n; u \in H^1(\Omega); \text{div} u = 0; u(x) = 0, x \in \partial\Omega\}.$$

3 Linear and nonlinear semigroups

3.1 Characterization of the Stokes operator and the semigroup it generates

Let us define the coercive and continuous bilinear form $a(\cdot,\cdot) : V \times V \rightarrow R$ as

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v dx.$$

Let $g \in H$. Then since $H \subset V' = \mathcal{L}(V;R)$, by Lax-Milgram lemma there exists $u \in V$ such that

$$a(u,v) = (g,v)_{L^2(\Omega)}, \quad \forall v \in V.$$

This defines the Stokes operator A as $Au = g$ with $u \in D(A)$ and $D(A) \subset V$ dense. In fact using the Cattabriga regularity theorem [2, 26, 24] we obtain an explicit representation of the domain of A as $D(A) = H^2(\Omega) \cap V$ and A is recognized as

$$Au = -P_H \Delta u, \quad \forall u \in D(A)$$

where $P_H : L^2(\Omega) \rightarrow H$ is the orthogonal projection. The Stokes operator is self adjoint and positive definite. Moreover $A \in \mathcal{L}(V;V') \cap \mathcal{L}(D(A);H)$ is an isomorphism on to. We have the continuous, dense and compact embeddings.

$$D(A) \subset V \subset H \equiv H' \subset V' \subset D(A)'.$$

Since A^{-1} is compact in H we can define the fractional powers $A^\alpha, \alpha \in R$ of A using its spectral resolution:

$$A^\alpha u = \int_0^\infty \lambda^\alpha dE(\lambda) u = \sum_1^\infty \mu_i^\alpha (u,\phi_i)_H \phi_i, \quad \forall u \in D(A),$$

where $E(\lambda)$ are the resolution of identity generated by A and $\{\mu_i, \phi_i\}$ are the eigen pair of A. The domain of A^α is

$$D(A^\alpha) = \{v = \sum_{i=1}^\infty a_i\phi_i; \|v\|^2_{D(A^\alpha)} = \sum_{i=1}^\infty \mu_i^{2\alpha} a_i^2 < \infty\}.$$

It follows from a theorem of Lions [15] that $D(A^{1/2}) = V$. We will denote by $D(A^{-\alpha})$ the dual of $D(A^\alpha)$ (closure of H under the norm of $D(A^{-\alpha})$). The resolvent of $-A$ satisfies

$$\|R(\lambda;-A)\|_{\mathcal{L}(H;H)} \leq \frac{1}{|\lambda + \epsilon|} \text{ for } \lambda \in \Sigma_{\lambda}, \quad 0 < \epsilon < \mu_1.$$

Here Σ_λ is a sector containing the right half plane.

Lemma 1 *$-A$ generates a* compact *semigroup that* is holomorphic *in Σ_A. Moreover for $0 \leq \beta \leq \alpha$,*

$$\|S(t)\|_{\mathcal{L}(D(A^\beta);D(A^\alpha))} \leq \frac{C}{t^{\alpha-\beta}}, \quad t > 0$$

and $\exists \epsilon > 0$ such that,

$$\|S(t)\|_{\mathcal{L}(H;H)} \leq e^{-\epsilon t}, \quad t \geq 0.$$

Here Σ_A is an acute sector containing the positive real axis.

3.2 Characterization of the inertia terms

Let us now consider the "inertia terms" in equation (1). We will define a linear operator L_U and bilinear operator $B(\cdot,\cdot)$ in the following way.

$$L_U v = P_H[(U \cdot \nabla)v + (v \cdot \nabla)U]$$

and $B(v,v) = P_H[(v \cdot \nabla)v]$.

Lemma 2 *If $U \in \Lambda$ then $L_U \in \mathcal{L}(D(A^\alpha);D(A^{\alpha-1/2})), \alpha \in [0,1/2]$. The bilinear operator satisfies $B(\cdot,\cdot) : V \times V \rightarrow D(A^{-1/4})$.*

Proof of these results can be found in [25]. The second result is actually due to [6].

3.3 The Linearized problem

Let us find $v \in C(0,\infty;V) \cap C^1(0,\infty;V')$ such that

$$v_t + L_U v + Av = 0, \quad t > 0.$$

$$v(0) = v_0 \in V.$$

This is resolved as

$$v(t) = S(t)v_0 - \int_0^t S(t-\tau)L_U(\tau)v(\tau)d\tau.$$

If we define the operator $\mathcal{K}$ as,

$$[\mathcal{K}v](t) = \int_0^t S(t-\tau)L_U(\tau)v(\tau)d\tau,$$

then $[(I + \mathcal{K})]v(t) = S(t)v_0$. Properties of $L_U(\cdot)$ and the estimates on $S(\cdot)$ allow us to show that for small enough T,

$$\|\mathcal{K}\|_{\mathcal{L}(C(0,T;V);C(0,T;V))} < 1.$$

Thus we have the convergent series representation.

$$v(t) = \sum_{n \geq 0} [\mathcal{K}^n S](t)v_0 = Z(t,0)v_0.$$

Uniqueness of the solution implies that the evolution operator $Z(t,\tau)$ satisfies $Z(t,\tau) = Z(t,\eta)Z(\eta,\tau), \; 0 \leq \tau \leq \eta \leq t$. Using this we extend $Z(t,\tau)$ to $0 \leq \tau \leq t < \infty$. Estimates on the Stokes semigroup allow us to prove,

Lemma 3 *For $\tau < t$ the maps $t \rightarrow Z(t,\tau)$ and $\tau \rightarrow Z(t,\tau)$ are continuous in the uniform operator topology of $\mathcal{L}(V;V) \cap \mathcal{L}(D(A^{-1/4});V)$. For $\tau < t$ the operator $Z(t,\tau) \in \mathcal{L}(V;V) \cap \mathcal{L}(D(A^{-1/4});V)$ is compact. We also have*

$$\|Z(t,\tau)\|_{\mathcal{L}(D(A^{-1/4});V)} \leq \frac{C}{(t-\tau)^{3/4}}, \quad t > \tau.$$

Let us now consider the case where the basic flow is *T-periodic* in time. Existence theorem for time periodic solutions for the Navier-Stokes equations can be found in [25]. When U is T-periodic, $Z(T,0)$ is called the monodromy operator and satisfies,

$$Z(nT,0) = Z(T,0)^n, n \geq 1.$$

3.4 Full Nonlinear problem

Let us find $v \in C(0,T^*(v_0);V)$ such that

$$v(t) = Z(t,0)v_0 - \int_0^t Z(t,\tau)B(v(\tau),v(\tau))d\tau$$

$$v(0) = v_0 \in V$$

where $T^*(v_0)$ the maximal time and is determined by the norm of the initial data v_0. We write $v(t) = W(t,0,v_0)$. Then we can show that [25].

$$v_0 \rightarrow W(t,0;v_0) \text{ is real analytic in } B_\delta(0) \subset V$$

This means $W(t,0,\cdot)$ can be written as a convergent power series

$$W(t,0;v_0) = \sum_{n\geq 1} \mathcal{H}_n(v_0,\cdots,v_0;t)$$

The n-linear maps $\mathcal{H}_n(\cdot,\cdots,\cdot;t)$ are continuous. We have

$$W(t,0;v_0) = W(t,\tau;W(\tau,0;v_0)), 0 \leq \tau \leq t < T^*(v_0).$$

For $\Omega \subset R^2$, we have $T^*(v_0) = \infty, \forall v_0 \in V$ [23]. For $\Omega \subset R^3$, $T^*(v_0) = \infty$ if v_0 is contained in a sufficiently small ball centered at the origin of V [6]. If U is T-periodic in time then $W(nT,0;\cdot) = W(T,0;\cdot)^n, \forall n \geq 1$. We note here that the Frechet derivative of this map $W(t,0;\cdot)$ is, $DW(t,\tau;0) = Z(t,\tau)$.

4 Invariant cones and manifolds

Let us first state the following fundamental result:

Theorem 1 *Let the spectrum of the monodromy operator $Z(T,0) \in \mathcal{L}(D(A^\alpha);D(A^\alpha))$ splits in to two disjoint sets σ_u and σ_s such that $\sigma(Z(T,0)) = \sigma_u \cup \sigma_s$ and*

$$b = \sup_{\sigma_s} |\lambda| < \inf_{\sigma_u} |\lambda| = a.$$

Let P_u and P_s be the spectral projectors defined by the Dunford's integrals,

$$P_u = \frac{1}{2\pi i}\int_{\Gamma_u} R(\lambda;Z(T,0))d\lambda \text{ and}$$

$$P_s = \frac{1}{2\pi i}\int_{\Gamma_s} R(\lambda;Z(T,0))d\lambda.$$

(Here $R(\lambda;Z(T,0))$ is the resolvent operator and Γ_u, Γ_s encircle σ_u, σ_s respectively). Then $\forall\epsilon > 0$, we can choose a norm in $D(A^\alpha)$, equivalent to the given one such that, $\forall v \in D(A^\alpha)$

$$(i)\ \ \|v\|_{D(A^\alpha)} = \|P_u v\|_{D(A^\alpha)} + \|P_s v\|_{D(A^\alpha)},\ \ \|P_u\| = \|P_s\| = 1,$$

$$(ii)\ \ \|Z(T,0)P_s v\|_{D(A^\alpha)} \leq (b+c)\|P_s v\|_{D(A^\alpha)},$$

$$(iii)\ \ \|Z(T,0)P_u v\|_{D(A^\alpha)} \geq (a-c)\|P_u v\|_{D(A^\alpha)}.$$

Here $P_s + P_u = I$, $P_s P_u = P_u P_s$ and P_s, P_u commute with $Z(T,0)$.

This result is in fact valid for any continuous linear operator in a Banach space with the above spectral properties. For the proof of this results see[4]. Let the spectrum of the monodromy operator splits into two sets such that $\sigma(Z(T,0)) = \sigma_u \cup \sigma_s$ with $\sigma_u \cap \sigma_s =$ empty and

$$b_s = \sup_{\lambda\in\sigma_s} |\lambda| < \inf_{\lambda\in\sigma_u} |\lambda| = b_u^{-1}$$

Theorem 2 (Invariant cone theorem) *If $b_u < 1$, then there exists a double cone that is invariant to $W(T,0,\ )$ locally. Let $\mathcal{K} = \{v \in V; \|P_s v\|_V \leq q\|P_u v\|_V, q > 0\}$ and let $\dot{\mathcal{K}} = \mathcal{K}\backslash\{0\}$. Then $W(T,0;\cdot) : B_\delta(0) \cap \dot{\mathcal{K}} \rightarrow \dot{\mathcal{K}}$ and is injective.*

This theorem is proved in [25] using a result on general mappings by Kirchgassner and Scheurle [10].

Theorem 3 The invariant manifold theorem Let $b_s, b_u < 1$. Then in a neighborhood $B_r(0) \subset D(A^\alpha), \alpha \in [1/2,1]$, there exists two unique, analytic manifolds M_s and M_u which are respectively the graphs of the analytic maps, $\phi_s : P_s D(A^\alpha) \rightarrow P_u D(A^\alpha)$, $\phi_u : P_u D(A^\alpha) \rightarrow P_s D(A^\alpha)$ with,

(*i*) $\phi_u(0) = \phi_s(0) = 0$,
(*ii*)$D\phi_u(0) = D\phi_s(0) = 0\cdots$tangency condition
(iii) manifolds M_s and M_u are locally invariant under the solution map $W(T,0;\cdot)$:

$$W(T,0;M_u \cap B_r(0)) \subset M_u \text{ and } W(T,0;M_s \cap B_r(0)) \subset M_s.$$

(iv) stable manifold M_s satisfies
$M_s \cap B_r(0) = \{u \in B_r(0)$ such that $\forall n \geq 0, W(nT,0;u) \in B_r(0)$ and $\rightarrow 0$ as $n \rightarrow \infty\}$

$$\text{and } M_s \cap B_r(0) \cap \dot{\mathcal{K}} = \textit{empty}$$

Unstable manifold M_u satisfies,
$M_u \cap B_r(0) = \{u \in B_r(0)$ such that $W(T,0,\cdot)^n u$ is defined $\forall n < 0$

$$\text{and tends to zero as } n \rightarrow -\infty\} \text{ and } M_u \cap B_r(0) \subset \mathcal{K}.$$

(v) if $u \notin M_s$ then there exists $\delta > 0$ and $p \in N$ such that, $\|W(pT,0;u)\|_{D(A^\alpha)} > \delta$.
(vi) $\text{dist}(M_u, W(T,0;u)) < \text{dist}(M_u,u)$ for $u \in B_r(0)\cdots$ exponential attractive property of the unstable manifold. $\text{dist}(M_s, W(T,0;u)) > \text{dist}(M_s,u)$ for $u \in B_r(0)\cdots$ repelling property of the stable manifold.

Proof of this result can be found in [25].

5 Regularized system: quest for global unique solvability

In this section we will present the results on a particular regularization of Navier-Stokes equations. For this system, it is possible to prove global unique solvability theorem up to dimension six [16],[18],[19]. In this regard the regularized system in six dimensions or less, behaves like the two dimensional Navier-Stokes equations.

Let $\Omega \subset R^n, n \leq 6$ be a smooth bounded domain. Find $(v^\epsilon, p^\epsilon) : \Omega \times (0,\infty) \rightarrow R^n \times R$ such that,

$$v^\epsilon_t + (v^\epsilon\cdot\nabla)v^\epsilon = -\nabla p^\epsilon + \nu\Delta v^\epsilon - \epsilon\Delta^2 v^\epsilon + f \qquad \text{in } \Omega\times(0,\infty),$$

$$\nabla\cdot v^\epsilon = 0 \qquad \text{in } \Omega \times (0,\infty), \tag{9}$$

$$v^\epsilon(x,t) = 0,\ \ (x,t) \in \partial\Omega \times (0,\infty),$$

$$\frac{\partial v^\epsilon}{\partial n}(x,t) = 0,\ \ (x,t) \in \partial\Omega \times (0,\infty)$$

and $v^\epsilon(x,0) = v^\epsilon_0 \qquad x \in \Omega.$

It is also possible to use periodic boundary condition (periodic in R^n) for this system. Let us define $\hat{V} := \{v \in H^2_0(\Omega); \nabla.v = 0\}$. Then as in the characterization of the Stokes operator, we can define a positive definite, selfadjoint operator $\hat{A} = P_H\Delta^2$ with $D(\hat{A}) = \hat{V} \cap H^4(\Omega)$ then $D(\hat{A}^{1/2}) = \hat{V}$ as before. We will obtain $B(\ ,\ ) . \hat{V} \times \hat{V} \rightarrow H$ compared to $B(\ ,\ ) . V \times V \rightarrow D(A^{-1/4})$. This shows that the nonlinearity in the regularized system has a better behavior. We can prove that,

Theorem 4 *If $\Omega \subset R^n$, $2 \leq n \leq 6$, then for a given $f \in L^2(0,T;H)$ and $v^\epsilon_0 \in H$, $\exists$ a unique solution v^ϵ to (9) such that, $v^\epsilon \in L^2(0,T : \hat{V}) \cap C([0,T];H)$.*

6 Global invariant manifolds

For the regularized system introduced in section (5) we can prove the existence of a compact global attractor as well as the local invariant manifolds of the type discussed earlier [19],[20]. Moreover, it is also possible to establish the existence of global invariant varieties of the type introduced in [5]. These global invariant varieties are modelled on finite dimensional linear manifolds spanned by the eigenfunctions of $\hat{A}$. The global invariant varieties defined below have C^1 smoothness as compared to the Lipschitz manifolds proposed in [5].

Definition 1 *$\mathcal{M}$ is called a C^1-*inertial manifold *if it is finite dimensional Lipschitz manifold whose derivatives are Lipschitz, has compact support, exponentially attractive, contains the global attractor and is invariant to the action of the solution map (the nonlinear semigroup) in the neighborhood of the global attractor.*

For the two dimensional regularized system with periodic boundary conditions we have

Theorem 5 *Let N^* be sufficiently large so that the eigenvalues of the operator $\hat{A}$ satisfy*

$$\mu_{N^*+1}^{1/2} \geq d_1/\epsilon \quad \textit{and} \quad \mu_{N^*+1}^{1/2} - \mu_{N^*}^{1/2} \geq d_2/\epsilon \qquad (10)$$

Then, $\forall N > N^*$, $\exists$ C^1-inertial manifolds $\mathcal{M}_N$ which are graph of maps from $P_N\hat{V}$ in to $Q_N\hat{V}$, where, $P_N = (I - Q_N)$ is the projection in-to the invariant subspace corresponding to the first N-eigenvalues of $\hat{A}$.

Proof of this theorem involves two steps. First we use the general method in [5] to establish the existence of Lipschitz-manifolds. The spectral condition 10 is in fact satisfied for this case. We then prove that under the same hypothesis of the theorem, the above manifolds have Lipschitz derivatives [19]. This is proven using methods similar to that in [25] for the analyticity theorem for the stable-unstable manifolds as described in the early part of this paper. These results on the regularized system motivate the Limit case $\epsilon \to 0$: Let v be the Hopf-class weak solution to the Navier-Stokes equations. Then for $\Omega \subset R^n, n = 2,3$ we have,

Theorem 6 *If $v^\epsilon \in L^\infty(0,T,L^4(\Omega))$ uniformly in ϵ then $\exists C > 0$ such that for $\nu > C$ we have as $\epsilon \to 0$, $v^\epsilon \to v$ strongly in $L^2(0,T;H)$*

Recall that. If the Hopf-class weak solution to the NS equations $v \in L^\infty(0,T;H) \cap L^2(0,T;V)$ satisfies $v \in L^\infty(0,T;L^4(\Omega))$ then it is unique.

7 Upper semicontinuous global attractors

We will now state the following powerful result [20] regarding such convergence for the two dimensional case. Let us define the global attractor for the regularized system (9) as

$$\Lambda_\epsilon = \bigcap_{\tau \geq 0} \overline{\bigcup_{t \geq \tau} W_\epsilon(t,0 : B_R(H))}$$

where $W_\epsilon(t,0:\cdot)$ is the nonlinear semigroup associated with the regularized system and $B_R(H)$ denotes absorbing ball in H for the semigroup $W_\epsilon(t,0;\cdot)$.

Theorem 7 *$\forall \epsilon \geq 0$ and $\forall \nu > 0$ the global attractor Λ_ϵ corresponding to the two dimensional regularized system (with periodic boundary conditions) has the following properties.*
(i) Λ_ϵ is compact and attracts boundes sets of H.
(ii) $W_\epsilon(t,0:\Lambda_\epsilon) = \Lambda_\epsilon$, $t \in R$.
(iii) Λ_ϵ is upper semicontinuous at $\epsilon = 0$: $\delta_H(\Lambda_\epsilon,\Lambda) \to 0$ as $\epsilon \to 0$. Here the semidistance δ_H is defined as

$$\delta_H(\Lambda_\epsilon,\Lambda) = \sup_{u\in\Lambda_\epsilon} \inf_{v\in\Lambda} \|u - v\|_H.$$

We note that Λ above corresponds to the global attractor for the two dimensional conventional Navier Stokes system.

Proof of this theorem involves certain uniform estimates (independent of ϵ) for the solution of the regularized system and an approximation result on the upper semicontinuous global attractors for nonlinear semigroups. The details of the proof can be found in [20].

Remark: In order to show the continuity of Λ_ϵ at $\epsilon = 0$, we need to prove the lower semicontinuity result: $\delta_H(\Lambda,\Lambda_\epsilon) \to 0$ as $\epsilon \to 0$. This issue is presently open.

Acknowledgement: This research has been supported by the faculty research innovation fund of the University of Southern California and by the Office of Naval Research through the URI Research Contract NO: N00014-86-K-0679.

References

[1] A.V. Babin and M.I. Vishik. Regular attractors of semi groups and evolution equations. *J. Math. Pure. Appl*, 62:441–491, 1983.

[2] L. Cattabriga. Su un problema al contorno relativo al sistema di equazioni di Stokes. *Rend. Mat. Sem. Univ. Padova*, 31:308–340, 1961.

[3] P. Constantin, C. Foias, and R. Temam. Attractor representing turbulent flows. *Memoirs Amer. Math. Soc.*, 53(314), 1985.

[4] M. A. Krasnosel'skii et al. *Approximate solutions of operator equations.* Moskou, 1969.

[5] C. Foias, G. R. Sell, and R. Temam. Inertial manifolds for nonlinear evolutionary equations. Preprint Series 234, IMA, University of Minnesota, March 1986.

[6] H. Fujita and T. Kato. On the Navier-Stokes initial value problem I. *Arch.Rat.Mech.Anal.*, 16(4).269–315, 1964.

[7] E. Hille and R.S.Phillips. *Functional analysis and semigroups.* 31. American mathematical society, Providence,R.I, 1957.

[8] M.W. Hirsch and C.C.Pugh. Stable manifolds and hyperbolic sets. In *Proceedings of the symposium in pure mathematics.* AMS, 1970.

[9] E. Hopf. Ein allgemeiner endlichkeitssatz der hydrodynamik. *Math. Ann.*, 117.764–775, 1941.

[10] K. Kirchgassner and J.Scheurle. Bifurcation. In J.K.Hale L.Cesari and J.P.LaSalle, editors, *Dynamical systems.* Academic press, 1976.

[11] O. A. Ladyzhenskaya. A dynamical system generated by the Navier-Stokes equations. *J. of Soviet Math*, 3:458–479, 1976.

[12] O. A. Ladyzhenskaya. On the finiteness of the dimension of bounded invariant sets for the Navier-Stokes equations and other related dissipative systems. In *Boundary value problems of mathematical physics and related questions in functional analysis*, volume 14. Steklev Institute, Leningrad, 1988.

[13] O.A. Ladyzhenskaya and V.A.Solonnikov. On the linearization principle and on invariant manifolds for problems of magnetohydrodynamics. *Zap.Nauch.Sem.Lomi. Leningrad*, 38:46-93, 1973.

[14] E. Lieb. On characteristic exponents in turbulence. *Comm. Math. Phys*, 92:473-480, 1984.

[15] J. L. Lions. Espaces d'interpolation et domaines de puissances fractionnaires d'operateurs. *J.Math.Soc.Japan*, 14(2):233-241, 1962.

[16] J. L. Lions. *Quelques méthodes de résolution des problémes aux limites non linéaires.* Dunod, Paris, 1969.

[17] R. Mane. On the dimension of the compact invariant sets of certain nonlinear maps. In *Lecture Notes in Mathematics., vol 898*, pages 230-242. Springer-Verlag, 1981.

[18] Y. R. Ou and S. S. Sritharan. Analysis of regularized Navier-Stokes equations-I. To be published, 1989.

[19] Y. R. Ou and S. S. Sritharan. Analysis of regularized Navier-Stokes equations-II. To be published. See also ICASE report 89-14, 1989.

[20] Y. R. Ou and S. S. Sritharan. Upper semicontinuous global attractors for viscous flow. To be published. See also ICASE Report 90-2, 1990.

[21] D. Ruelle. Large volume limit of distribution of characteristic exponents in turbulence. *Comm. Math. Phys*, 87:287-302, 1982.

[22] D. Ruelle. Characteristic exponents for viscous fluid subjected to time dependent forces. *Comm. Math. Phys*, 92:285-300, 1984.

[23] P. Sobolevskii. On the nonstationary equations of the hydrodynamics of a viscous fluid. *Dokl. Akad. Nauk SSSR*, 128:45-48, 1959.

[24] V. A. Solonnikov. Estimates of the Green tensor for some boundary problems. *Dokl. Akad. Nauk. SSSR*, 130:988-991, 1960.

[25] S. S. Sritharan. *Invariant Manifold Theory For Hydrodynamic Transition.* John Wiley, New York, 1990.

[26] I. I. Vorovich and V. I. Yudovich. Stationary flows of incompressible viscous fluids. *Mat. Sb.*, 53:393-428, 1961.

PRESENTATION OF A SECOND ORDER TIME SCHEME USING THE CHARACTERISTICS METHOD AND APPLIED TO THE NAVIER-STOKES EQUATIONS

K. BOUKIR, B. METIVET,
E. RAZAFINDRAKOTO
Electricité de France
Etudes et Recherches, IMA
1, av. du Général de Gaulle
92141 Clamart-Cedex France

and

Y. MADAY
Analyse Numérique
Université P. & M. Curie
Tour 55 - 65, 5e étage
4 Place Jussieu
75252 Paris Cedex 05 France

ABSTRACT - We present here an approximation of incompressible 2D or 3D Navier-Stokes equations in velocity-pressure formulation with a scheme which is second order in time and of finite element type in space. The convective part is treated according to a characteristics method, the Stokes part is solved with the help of an Uzawa algorithm.
Theoretical results concerning the precision and the stability are given . Numerical tests show the improvement given by this second order scheme compared to a first order scheme also using characteristics to treat the convective part.

I. DESCRIPTION OF THE SECOND ORDER TIME SCHEME

This scheme is based on an operator splitting method similar to the one proposed in [1], [2] and [3].

We denote by $\Omega \subseteq R^N$ (N = 2 or 3) the computational domain and by [0, T] the time interval. We consider the Navier-Stokes equations: find $v : \Omega \rightarrow R^N$ and $p : \Omega \rightarrow R$ solutions of

$$\frac{\partial v}{\partial t} + v \,.\nabla v - \nu \,\Delta v + \nabla p = f \quad \text{in } \Omega \times]0, T[, \qquad (1)$$

$$\operatorname{div} v = 0 \quad \text{in } \Omega \times]0, T[, \qquad (2)$$

$$v(x, 0) = v_0(x) \quad \text{for } x \in \Omega , \qquad (3)$$

$$v = v_d \quad \text{on } \Gamma \times]0, T[. \qquad (4)$$

Let Δt be the time step and $t^{n+1} = (n+1)\Delta t$. The second order time scheme is defined by two steps.

Concerning the convection step, for any point $x \in \Omega$, we introduce the characteristic curve $X_x^{n+1} : [t^{n-1}, t^{n+1}] \rightarrow R^N$, solution of

$$\begin{cases} \forall t \in [t^{n-1}, t^{n+1}[, \quad \dfrac{dX_x^{n+1}}{dt}(t) = \\ \qquad\qquad v^{n*}\,(X_x^{n+1}(t)), \text{ if } X_x^{n+1}(t) \in \overline{\Omega}, \\ \qquad\qquad 0, \qquad\qquad \text{otherwise}, \\ X_x^{n+1}(t^{n+1}) = x , \end{cases} \qquad (5)$$

where $v^{n*} = 2\, v^n - v^{n-1}$.

We set

$$\tilde{v}_1^{n+1}(x) = v^{n*}\,(X_x^{n+1}(t^n)) ,$$

and

$$\tilde{v}_2^{n+1}(x) = v^{n*}\,(X_x^{n+1}(t^{n-1})) .$$

The Stokes step consists in computing the approximations $v^{n+1} : \Omega \rightarrow R^N$ and $p^{n+1} : \Omega \rightarrow R$ of the velocity and the pressure at time t^{n+1}, solutions of

$$\begin{cases} \dfrac{\frac{3}{2} v^{n+1} - 2\,\tilde{v}_1^{n+1} + \frac{1}{2}\tilde{v}_2^{n+1}}{\Delta t} - \nu\,\Delta v^{n+1} + \operatorname{grad} p^{n+1} \\ \qquad\qquad = f(\,.\,, t^{n+1}) \quad \text{in } \Omega, \\ \operatorname{div} v^{n+1} = 0 \qquad \text{in } \Omega, \\ v^{n+1} = v_d\,(., t^{n+1}) \qquad \text{on } \Gamma. \end{cases} \qquad (6)$$

We denote by $\|\;\|_\nu$ the norm of $H^1(\Omega)$, defined by

$$\| u \|_\nu = [\, \| u \|_0^2 + \nu\,\Delta t\, \| \nabla u \|_0^2 \,]^{\frac{1}{2}}.$$

II. THEORETICAL RESULTS

In this section, we only consider problems (1)-(4) the solution (v, p) of which is sufficiently smooth. Moreover, we assume that $v_d = 0$.

II.1 Time consistency error

It is easy to see that only the momentum conservation equation produces a non zero consistency error.

For any point $x \in \Omega$, let $\chi_x^{n+1} : [t^{n-1}, t^{n+1}] \rightarrow R^N$ denote the characteristic curve defined by a system analogous to (5), where the discrete velocity is replaced by v. We also set

$$h_x(t) = v\,(\chi_x^{n+1}(t), t) \quad \text{for } t \in [t^{n-1}, t^{n+1}].$$

The consistency error is of order 2: it is equal to

$$E(v, p) = \operatorname{Sup} \{\, \| e(x, n+1) \|, \ x \in \Omega , n / t^{n+1} \in [0, T] \,\},$$

where

$$e(x, n+1) = - (\Delta t)^2 \Big[\frac{1}{3}\, \frac{d^3 h_x}{dt^3}(t^{n+1}) + \frac{\partial^2 v}{\partial t^2} .\nabla v\,(x, t^{n+1})\Big] + O([\Delta t]^3)$$

($\| \,.\, \|$ is the euclidian norm of R^N).

II.2 Stability results

The present results are concerned with unsteady flow computations — the steady case is under consideration —.
The case of the linear convection-diffusion problem

$$\frac{\partial v}{\partial t} + u \,.\nabla v - \nu\,\Delta v = f \quad \text{in } \Omega \times]0, T[,$$

$$v(x, 0) = v_0(x) \text{ for } x \in \Omega ,\ v = 0 \text{ on } \Gamma \times]0, T[, \qquad (7)$$

has been already studied in [1]. We can deduce from Ewing and Russel's convergence results that the scheme is unconditionally stable for the norm $\|\;\|_1$; moreover we have exhibited an upperbound of the sequence $(\| v^n \|_\nu)_n$ by using a somewhat different proof:

Proposition 1: *If* div u = 0 *in* $\Omega \times]0, T[$ *and* u.n = 0 *on* $\Gamma \times]0, T[$, *there exist constants* C (u), C (f), C (v^0, v^1), *depending only on* Ω *and respectively* u, f, (v^0, v^1), *such that for* $\Delta t \leq C(u)$, *we have*

$$\| v^n \|_V \leq C(f)\,T + C(v^0, v^1). \qquad (8)$$

In the nonlinear case of the Navier-Stokes equations, we can presently get stability results only when introducing the space discretization. Let h be the space step and k the degree of the velocity finite elements. By using technics similar to those of [1], we can proove the

Proposition 2: *There exist constants* C_1 *and* C_2 *depending only on* Ω *and respectively* (v, v_h^0, v_h^1) *and* $(v, v_h^0, v_h^1, e^{\frac{TC_1}{2v}})$ *such that, for* h *small enough and* k > N/2, *the condition* $\Delta t \leq \inf(1/C_1, C_2 h^{N/4})$ *yields to*

$$\| v_h^n \|_{1,\infty} \leq C_1. \qquad (9)$$

The scheme appears slightly less stable than the first order scheme, which is unconditionally stable ([6]), but this instability may be only due to the technics of proof.

III. NUMERICAL RESULTS

The second order time scheme has been implemented in the thermalhydraulic finite element code N3S developed at EDF ([4]). Comparisons have been done with the original first order scheme ([4], [5], [6]). In all the cases, the second order scheme improves the results ([7]).

III.1 Results on analytical steady cases

We have observed on analytical steady tests that the consistency error affects more the pressure results than the velocity ones (cf. Fig. 1 to 3 where results are shown for the same mesh and $\Delta t = 10^{-3}$).

However, for the tested cases, with the first (resp. second) order scheme, the computed time precision is of order 1 (resp. 2) for both velocity and pressure. We get thus a numerical information concerning the pressure rate of convergence since presently there is no theoretical results for any of the two schemes.

Moreover we have observed that a too strong refinement of the time step leads to worth results. In particular, when in the momentum equation the convective part is very important compared with the diffusion-pressure part, we cannot get a good pressure without refining the mesh together with the time step. The theoretical space-time precision of the second order scheme applied to the Navier-Stokes equations is under consideration (see [6] for the first order one).

III.2 Results on an unsteady case

The computations of the Gamm workshop ([8]) have been performed with the two schemes. They are concerned with a flow in a heated cavity at a zero Prandtl number and with homogeneous Dirichlet boundary conditions. The following result is given as an example of the improvements given by the second order scheme. For a Grashof number equal to 30000, the reference computations predict an unsteady periodic state which is well computed with the second order scheme, although the first order one converges to a steady state.

IV. GENERALIZATION

This scheme can be generalized to a k-order time scheme ([3]). The consistency error has been studied in [7]. Concerning the order 3, the scheme has been succesfully used together with the spectral methods ([2]). The stability results are under consideration.

References

[1] R. E. Ewing, T. F. Russel, *"Multistep Galerkin Methods along Characteristics for Convection-Diffusion Problems."*, Advances in Comp. Meth. for P.D.E., R. Vichnevetsky & R.S. Stepleman eds., IMACS, Rutgers Univ., New Brunswick, N.J., 1981, pp 28-36.

[2] L Ho, Y. Maday, A. Patera, E. Ronquist, *"A high order Lagrangien decoupling method for the incompressible Navier-Stokes equations."* Proceedings of ICOSAHOM'89 meeting, C.Canuto & A. Quarteroni eds, North Holland (1990).

[3] Y. Maday, A. Patera, E. Ronquist, *"An operator integration factor splitting method for time dependant problems. Application to incompressible fluid flows."* To appear in Journal of Scientific Computing.

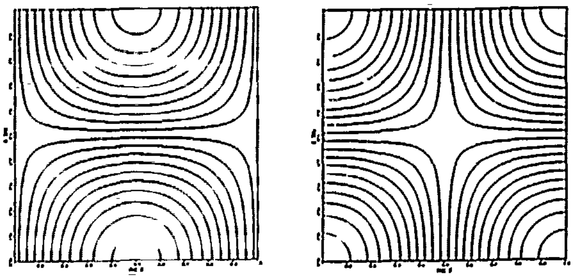
[4] J.P. Chabard, *"N3S code for fluid mechanics –theoretical manual – release 2.0."* EDF Report ref.HE41/89.14 (1989).

[5] J.P. Benque, B. Ibler, A. Keramsi, G. Labadie, *"A finite element method for the Navier-Stokes equations."*, Proceedings of the third international conference on finite elements in flow problems. Banff.Alberta, Canada, 10-13 June 1980.

[6] O. Pironneau, *"On the transport diffusion algorithm and its applications to the Navier Stokes equations."* , Numer. Math. 38, 309-332, 1982.

[7] B. Métivet, E. Razafindrakoto, *"Projet N3S de mécanique des fluides. Etude numérique d'un schéma aux caractéristiques d'ordre 2 pour la résolution des équations de Navier-Stokes."* EDF Report ref. HI72/7094 (1990).

[8] Numerical simulation of oscillatory convection in low-Pr fluids, GAMM Workshop, Bernard Roux Ed., Marseille, 1989.



Isolines of v_x Iso-pressure lines

Figure 1: analytical solution

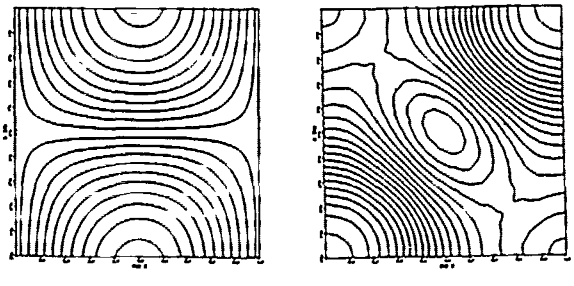


Figure 2 : solution computed with the first order scheme

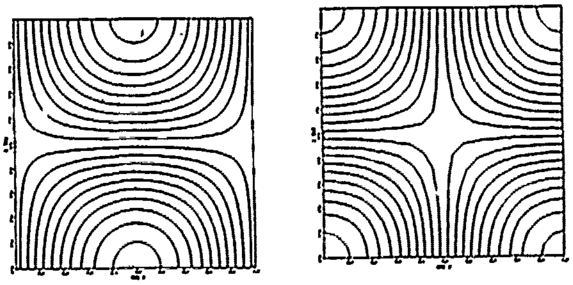


Figure 3 . solution computed with the second order scheme

HOW DOES THE NAVIER-STOKES EQUATION ENGENDER CHAOS?

JON LEE
Flight Dynamics Dir. (FIB)
Wright-Patterson AFB, OH 45433 USA

Abstract - *For the 2D Navier-Stokes equations as a specific model for dynamical systems, we have found that the chaotic transition is couched on a 2-torus-like manifold which is the product space of a circle and multiply-periodic orbit. This is conceptually similar to the unstable 2-torus of Ruelle-Takens-Newhouse.*

I. INTRODUCTION

Although it is agreed that the laminar-to-turbulence transition begins with emergence of a periodic motion from the fixed point (laminar flow), there is no general agreement as to what happends next to the periodic motion as the forcing (Reynolds number) is further increased. Several scenarios leading to chaos have been proposed;the unstable quasiperiodic 3-torus of Ruelle-Takens-Newhouse[1,2], period-doubling of Feigenbaum[3], and intermittency of Pomeau-Manneville [4] (see, for instance, the review article of Eckmann [5]).We shall show that the 2D Navier-Stokes equations in a cyclic domain follow, at least conceptually, the scenario of Ruelle-Takens-Newhouse closer than any others, although the actual state of affairs is somewhat more complicated. Prior to chaos, there appears a multiply-periodic orbit and the chaotic transition takes place on a 2-torus-like manifold which is the product space of a circle along the longitude angle and the multiply-periodic orbit in the plane of latitude angle. Instead of the perturbation of a circle map[6,7], it therefore appears that the appropriate model for chaotic transition would be the area-preserving map of Chirikov[8] generalized to include anharmonicity.

II. A SET OF 860 EVOLUTION EQUATIONS

For a 2D periodic flow with no mean flow, it is most expedient to Fourier analyze the velocity field $\vec{U}(\vec{x})$ and body forces $\vec{G}(\vec{x})$ in a square region of side L by $\begin{pmatrix}\vec{U}(\vec{x})\\ \vec{G}(\vec{x})\end{pmatrix} = \sum_{\vec{k}} \begin{pmatrix}\vec{U}(\vec{k})\\ \vec{G}(\vec{k})\end{pmatrix} \exp(i\vec{k}\cdot\vec{x})$, where $\vec{k}=(2\pi/L)\begin{bmatrix}n_x\\ n_y\end{bmatrix}$ $(n_x, n_y=0,\pm1,\pm2,\dots)$ is the wavevector. By spanning incompressible $\vec{U}(\vec{k})$ and $\vec{G}(\vec{k})$ by the unit polarization vector $\vec{\varepsilon}(\vec{k})$ normal to $\vec{k}$, the incompressible Navier-Stokes equations give rise to the so-called triad-interaction representation in spectral form

$$\frac{\partial u(\vec{k})}{\partial t} + \nu k^2 u(\vec{k}) = -i \sum_{\vec{k}+\vec{p}+\vec{q}=0} \bar{\phi}_{\vec{k}|\vec{p},\vec{q}}\, u^*(\vec{p})u^*(\vec{q}) + g(\vec{k}), \qquad (1)$$

where ν is the kinematic viscosity, $k=|\vec{k}|^{1/2}$ and $\bar{\phi}_{\vec{k}|\vec{p},\vec{q}}=(\vec{k}\cdot\vec{\varepsilon}(\vec{p}))(\vec{\varepsilon}(\vec{k})\cdot\vec{\varepsilon}(\vec{q}))+(\vec{k}\cdot\vec{\varepsilon}(\vec{q}))(\vec{\varepsilon}(\vec{k})\cdot\vec{\varepsilon}(\vec{p}))$ is the symmetrized coupling coefficient. Let us label the wavevectors in the successive k-rings. There are 430 wavevectors for isotropic truncation of the upper wavenumber K=16. By abbreviating $u(\vec{k}_n)=u_n$ and $g(\vec{k}_n)=g_n$, enumeration of (1) yields a set of 430 equations for u_n with 106,244 triad-interaction terms in the right hand side.

For computation we split u_n into the real and imaginary parts by $u_n=u_n^r+iu_n^i$ and, similarly, $g_n=g_n^r+ig_n^i$. Denote the vector of u_n^r and u_n^i by $\mathfrak{U}$, $k_n^2u_n^r$ and $k_n^2u_n^i$ by $\mathcal{F}_0(\mathfrak{U})$, the triad-interactions by $\mathcal{F}_1(\mathfrak{U},\mathfrak{U})$, g_n^r and g_n^i by $\mathfrak{G}$. The set of 860 equations has the vector form

$$\dot{\mathfrak{U}} = -\nu\mathcal{F}_0(\mathfrak{U}) + \mathcal{F}_1(\mathfrak{U},\mathfrak{U}) + \mathfrak{G}, \qquad (2)$$

similar to the evolution equation of Constantin et. al.[9]. The energy and enstrophy conservations follow from $\langle\mathfrak{U},\mathcal{F}_1\rangle=\langle\mathcal{F}_0,\mathcal{F}_1\rangle=0$,where <,> is the scalar product.

III. FIVE EQUILIBRIUM STATES

For the present study, we shall restrict ourselves to the single-mode and three-mode forcing

$$G_1 = (0 \;\dots\; g_{26}^r, g_{26}^i, \dots\; 0),$$

$$G_3 = (0 \;..\; g_{26}^r, g_{26}^i, g_{27}^r, g_{27}^i, g_{28}^r, g_{28}^i, \dots\; 0),$$

where g's are assumed constant. For the time independent forcing, trajectory behavior can best be categorized by the amplitude-angle form $u_n=R_n\exp(i\omega_n)$. Hence, the equations of motion in R_n and ω_n

$$\dot{\mathcal{V}}_r = \mathcal{H}_r(\mathcal{V}_r,\mathcal{V}_\omega),\; \dot{\mathcal{V}}_\omega = \mathcal{H}_\omega(\mathcal{V}_r,\mathcal{V}_\omega). \qquad (3)$$

represent an assembly of 430 coupled u_n-oscillators. Here, $\mathcal{V}_r$ and $\mathcal{V}_\omega$ are the vectors of R_n and ω_n, $\mathcal{H}_r$ and $\mathcal{H}_\omega$ are the right-hand sides involving the forcing $g_j=F_j\exp(i\theta_j)$ for j=26, 27 and 28.

We have found experimentally that (3) admits the following equilibrium;

$$\bar{R}_n\text{=const},\; \bar{R}_n\text{=periodic},\; \bar{R}_n\text{=chaotic}, \qquad (4a,b,c)$$

and $\bar{\omega}_n$=const, $\bar{\omega}_n$=const x t, $\bar{\omega}_n$=periodic,

$$\bar{\omega}_n\text{=const x t + periodic},\; \bar{\omega}_n\text{=chaotic}. \qquad (5a\text{-}d)$$

Only certain combinations of (4) and (5) give rise to to the equilibrium states (ES) as follows;

ES I: Fixed Point - eqn (4a) & (5a)
ES II: Circle - eqn (4a) & (5b)
ES III: Closed orbit - eqn (4b) & (5c)
ES IV: Torus - eqn (4b) & (5d)
ES V: Chaos - eqn (4c) & (5e)

For ES III the principal frequencies of $\bar{R}_n$ and $\bar{\omega}_n$ must be rationally related, otherwise the orbit is unclosed. What is, however, unexpected is ES IV. Let us denote the equilibrium state by $X=R\exp(i\Omega)$ without the overhead bar. The combination of (4b) and (5d) is explicitly given by

$$R=R_0+\Delta r\sin(2\pi f_r t+\varphi_r),\; \Omega=\Omega_0+\Omega' t+\Delta\omega\cos(2\pi f_\omega t+\varphi_\omega). \qquad (6)$$

Here, R is modulated by a sine with amplitude Δr, frequency f_r, phase φ_r, and $\Delta\omega$, f_ω, φ_ω are defined similarly for Ω. The angular velocity Ω' is measured positive/negative for counter-clockwise/clockwise rotation. Since $\Omega' T=2\pi(0.16)$ in Fig 1(a), the phase plot of X^r vs X^i has rotated counterclockwise by about 58° in Fig 1(b). Note that $R=R_0$ and $\Omega=\Omega_0+\Omega' t$ is a circle S^1 of radius R_0(ES II), and $R=\Delta r\sin(2\pi f_r t+\varphi_r)$ and $\Omega=\Delta\omega\cos(2\pi f_\omega t+\varphi_\omega)$ form a closed curve C^1(ES III).

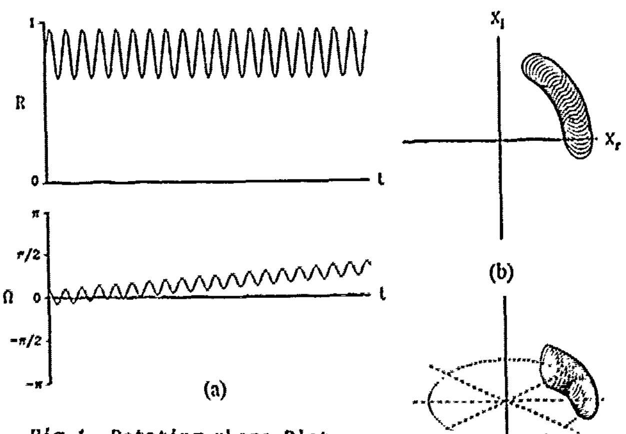


Fig 1. Rotating phase Plot.
(a) $R=0.8+0.15\sin2\pi(0.012t-0.3)$,
$\Omega\; 2\pi(0.0001t)+0.04\cos2\pi(0.012t)$
over T=1600; (b) Phase plot;
(c) Construction of 2-torus.

Hence, (6) represents a 2-torus-like manifold(ES IV) of the product space $C^1 x S^1$. For definiteness, we shall say latitude angle $\Omega_1 (=2\pi f_\omega)$ encloses C^1 and longitude angle $\Omega_2 (=\Omega')$ revolves C^1 around the S^1(see, Fig 165 of Arnold[10]). By decoupling the two angles, one can generate a segment of 2-torus as shown in Fig 1(c).

VI. NUMERICAL EVIDENCE FOR CHAOTIC TRANSITION

A large $\nu = 0.02$ is used to assure the validity of isotropic truncation for K=16. Since $F_{26}=F_{27}=F_{28}\equiv F$, forcing amplitude F is the parameter for trajectory behavior. Numerical experiments have shown that ES III evolves into ES V within a very small range of F. Since u_n-oscillators are dynamically similar, we shall pick out u_{132}-oscillator and examine how chaos can develop out of a multiply-periodic orbit.

Under G_1 the chaotic transition takes place in F=(0.1601,0.162). First, we present in Fig 2(a) the phase plot at F=0.1601, which is closed. Fig 2(b) shows that R_{132} and ω_{132} are periodic. Now, as F is raised to 0.1615, we find in Fig 3(a) that the phase plot is no longer closed, but appears rotating about the origin of the phase plane. From the angle of Fig 3(b), one can estimate longitude angle $\Omega' \approx -2\pi(0.000055)$ and latitude angle $2\pi f_\omega \approx 2\pi(0.0079)$ by (6). Hence, the orbit of u_{132}-oscillator is on a 2-torus-like manifold(ES IV) which is a wrinkly dough nut with contorted but smooth cross-sections. For F> 0.1615, the orbit cannot remain on ES IV, hence is attracted to a strange attractor.

Similarly, under G_3 the chaotic transition takes place in F=(0.06722, 0.06729). Here, the existence of ES IV is found almost at a point value of F. As F increases toward F= 0.06729, we shall come to pass a threshold value (yet undetermined), beyond which u_{132}-oscillator evolves continually from a multiply periodic orbit to chaos. To be specific, we have shown in Fig 4 a sequence of three phase plots observed at F=0.06727. First, the orbit splits into two loops(Fig 4(a)) and then each loop overlaps in the phase plane (Fig 4(b)). Note that separation is brought about by the period-doubling of R_{132} and ω_{132}, and the overlapping is due to the longitude angle. Finally, there comes the climactic phase plot of Fig 4(c) which can no longer restrain the orbit on ES IV, thereby thrusting into a strange attractor.

V. Conclusions

One can schematize the chaotic transition as shown in Fig 5. Prior to chaos, a multiply-periodic orbit lies on the plane of latitude angle. However, with the emergence of longitude angle there forms a 2-torus-like manifold, similar to the quasiperiodic 2-torus of Ruelle & Takens[1] (see, Fig 1.14 of Ref [11]). Hence, a generalization of Chirikov's[8] area preserving map may be appropriate for ES IV

$$R_{i+1}=R_i-\frac{\rho}{2\pi}[\sin 2\pi\theta_i+\sigma_2 \sin 4\pi\theta_i \; ..], \quad \theta_{i+1}=\theta_i+\beta+R_{i+1}. \quad (7)$$

Here, we indicate only the second harmonic of strength σ_2, in the absence of which (7) is the original map of Chirikov[8]. It further reduces to the circle map $\Phi(\theta) =\theta+\beta\frac{\rho}{2\pi}\sin 2\pi\theta$ that Shenker[6] and Rand et.al.[7] have investigated as a model for chaotic transition of quasiperiodic orbit on a 2-torus.

References

1. Ruelle,D. & F.Takens,Comm. Math. Phys. *20*,167(1971)
2. Newhouse,S.,D.Ruelle & F.Takens, Comm. Math. Phys., *64*, 35 (1978).
3. Feigenbaum,M.J., J. Stat. Phys., *19*,25 (1978).
4. Pomeau,Y. & P.Manneville, Comm. Math. Phys., *77*, 189 (1980).
5. Eckmann,J.-P., Rev. Mod. Phys., *53*, 643 (1981).
6. Shenker,S.J., Physica *5D*, 405 (1982).
7. Rand,D., S.Ostlund, J.Sethna & E.D.Siggia, Phys. Rev. Lett., *49*, 132 (1982).
8. Chirikov,B.V., Phys. Rept., *52*, 265 (1979).
9. Constantin,P., C.Foias, O.P.Manley & R.Temam, J. Fluid Mech., *150*, 427 (1985).
10. Arnold,V.I., *Ordinary Differential Equations*, The MIT Press (1980).
11. Marsden,J.E. & M.McCracken, *The Hopf Bifurcation and its Applications*, Appl. Math. Sci. *19*, Springer-Verlag (1976).

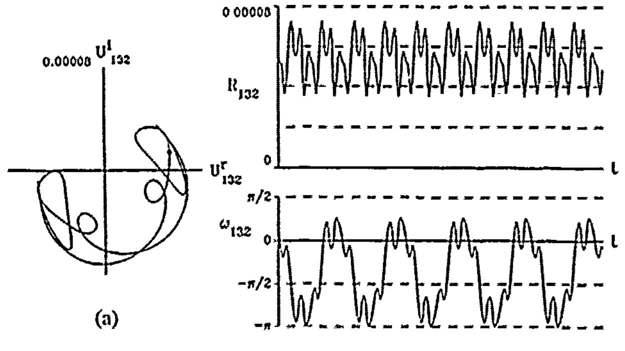


Fig 2. The u_{132}-oscillator under G_1 with F=0.1601. (a) Phase plot; (b) Amplitude and angle over T 660.

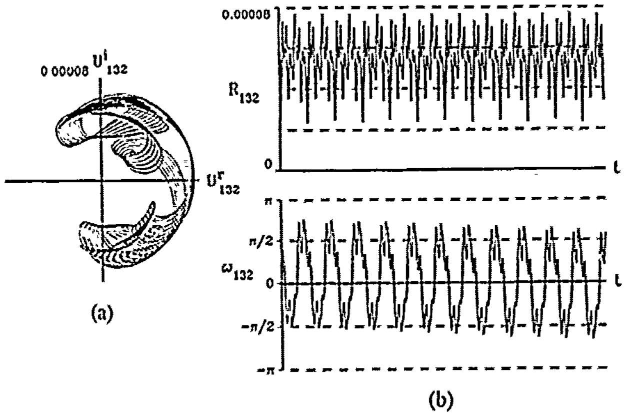


Fig 3. The u_{132}-oscillator under G_1 with F=0.1615. (a) Phase plot; (b) Amplitude and angle over T-1500.

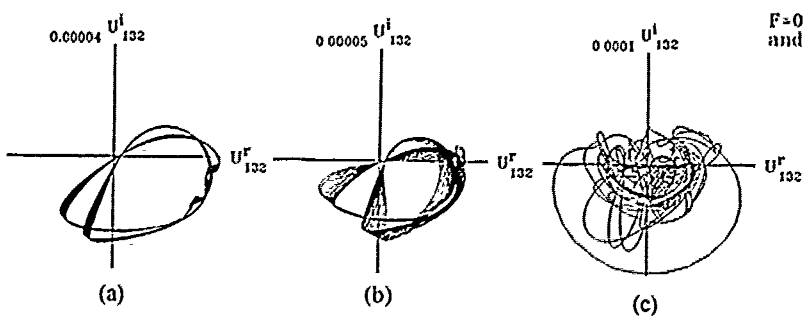


Fig 4. Phase plots under G_3 with F 0 6727. (a) T 9600; (b) T-10800, (c) T-12000

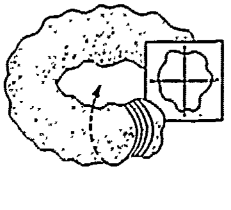


Fig 5. Schematic view of chaotic transition.

THE STABILITY OF DIFFERENCE SCHEMES OF A PARABOLIC EQUATION

Sun Qiren
Shanghai Institute of Applied Math. & Mech.
149 Yanchang Road, Shanghai, 200072
People's Republic of China

Abstract— This paper proposes a new method to improve the stability condition of difference schemes of a parabolic equation. Necessary and sufficient conditions of the stability of this new method are given and proved. Some numerical examples show that this method has some calculation advantages.

I. Introduction

For the parabolic equation,

$$\begin{cases} \frac{\partial u}{\partial t} = a\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f(x,y,z,t), & (x,y,z)\in\Omega,\ o < T \leqslant T \\ u(x,y,z,t) = o, & (x,y,z)\in\Gamma,\ o < t \leqslant T \\ u(x,y,z,o) = \varphi(x,y,z), & (x,y,z)\in\Omega \end{cases} \tag{1}$$

where

$$\Omega = \{(x,y,z) \mid 0 < x < 1,\ 0 < y < 1, 0 < z < 1\}, \quad \Gamma = \partial\Omega,$$

a is a positive constant, many feasible difference schemes were constructed on difference computation of the initial and boundary problem by many authors. When we use and explicit difference sheeme to computate the problem (1), the step is limited greatly in order to satisfy stability conditions.

This paper proposes a new method that a fixed space step and alternative time steps are adopted in an arbitray choosing explicit difference scheme to improve stability conditions for explicit difference schemes which are used to solve the parabolic equation (1.1). According to general idea, stability conditions of the explicit scheme should be satisfied on each time-level. As a matter of fact, it is not necessary. When we adopt suitable stable and unstable schemes alternately, then it finally leads to a stable scheme. In this paper we give and prove the improved stablity conditions which are much better than the stability conditions of the classical explicit scheme. Therefore, we can increase the time step in practical calulation, finish calculation work in less time and bring calculation advantages of explicit schemes into full play.

II. Sufficient and Necessary Condition of Stability

We can use a lot of kinds of explicit schemes to calculate the equation (1). Here we use the simplest classical explicit difference scheme to calculate the problem (1), and then improve stability conditions step by step.

The classical explicit difference scheme of the equation (1) is

$$\begin{cases} \frac{U^{k+1}_{i,j,l} - U^{k}_{i,j,l}}{\tau} = \frac{a}{h^2}(U^k_{i+1,j,l} + U^k_{i-1,j,l} + U^k_{i,j+1,l} + U^k_{i,j-1,l} \\ + U^k_{i,j,l+1} + U^k_{i,j,l-1} - 6U^k_{i,j,l}) + f^k_{i,j,l}, \quad i,j,l = 1,\cdots,N-1 \\ U^k_{0,j,l} = U^k_{N,j,l} = U^k_{i,0,l} + U^k_{i,N,l} = U^k_{i,j,0} = U^k_{i,j,N} = 0 \\ U^0_{i,j,l} = \varphi_{i,j,l}, \quad i,j,l = 0,\cdots, N \end{cases} \tag{2}$$

where τ is a time step, $h = 1/N$ is a space step, N is a positive integer number, $f^k_{i,j,l} = f(ih,jh,lh,k\tau)$, $\varphi_{i,j,l} = \varphi(ih,jh,lh)$. Set $r = a\tau/h^2$ We casily have that the stability condition of the scheme of (2) is

$$r \leqslant \frac{1}{8} \tag{3}$$

Then we consider alternative explicit difference scheme of the

the

problem (1).

$$\begin{cases} \frac{U^{k+1}_{i,j,l} - U^{k}_{i,j,l}}{\tau} = \frac{a}{h^2}(U^k_{i+1,j,l} + U^k_{i-1,j,l} + U^k_{i,j+1,l} + U^k_{i,j-1,l} \\ + U^k_{i,j,l+1} + U^k_{i,j,l-1} - 6U^k_{i,j,l}) + f^k_{i,j,l}, \quad i+j+l = odd\ number. \\ \frac{U^{k+1}_{i,j,l} - U^{k}_{i,j,l}}{\tau} = \frac{a}{h^2}(U^{k+1}_{i+1,j,l} + U^{k+1}_{i-1,j,l} + U^{k+1}_{i,j+1,l} + U^{k+1}_{i,j-1,l} \\ + U^{k+1}_{i,j,l+1} + U^{k+1}_{i,j,l-1} - 6U^{k+1}_{i,j,l}) + f^k_{i,j,l}, \quad i+j+l = even\ number. \\ U^k_{0,j,l} = U^k_{N,j,l} = U^k_{i,0,l} + U^k_{i,N,l} = U^k_{i,j,0} = U^k_{i,j,N} = 0 \\ U^0_{i,j,l} = \varphi_{i,j,l}, \quad i,j,l = 0,\cdots, N \end{cases} \tag{4}$$

Set $u^k_{i,j,l} = \zeta^k_1 e^{i(\sigma_1 i + \sigma_2 j + \sigma_3 l)h}$, when i+j+l is a odd number,

$u^k_{i,j,l} = \zeta^k_2 e^{i(\sigma_1 i + \sigma_2 j + \sigma_3 l)h}$, when i+j+l is an even nuber,

and use separated variable method to find a stability condition of scheme (4) The sufficient and necessary condition of scheme (4) is

$$r \leqslant \frac{1}{3} \tag{5}$$

Finally, in this paper we adopt a fixed space step h and use time steps τ_2 and τ_1 alternately. If we use τ_2 in odd time levels and τ_1 in even time levels and two levels are considered as a whole, the following conclusion for improving stability conditions can be obtained.

Theorem 1. To the equation (1), when the time steps τ_2 and τ_1 are alternately used to difference scheme (4), the sufficitent and necessary conditions of steability are

$$(I)\begin{cases} R_1 \geqslant 6R_2 \\ R_1 \geqslant 2\sqrt{R_2} \\ R_1^4 - 16R_2^2 - 576R_2^3 \leqslant 0 \end{cases} \qquad and(II)\begin{cases} 3R_1 \leqslant 18R_2 + 1 \\ R_1 \geqslant 6\sqrt{2}R_2 \\ R_1^4 - 16R_2^2 - 576R_2^3 \geqslant 0 \end{cases} \tag{6}$$

where $r_1 = a\tau_1/h^2$, $r_2 = a\tau_2/h^2$, $R_1 = r_1 + r_2$, $R_2 = r_1 r_2$

Proof. First, we prove that the sum of the areas (I) and (II) is the necessary stability conditions.

When the time steps τ_2 and τ_1 are alternately used to the homogeneous scheme of difference scheme (4) and two levels are combined into a whole, a new augmented matrix is

$$G = \begin{bmatrix} 1-6r_1 & 2r_1 C \\ \frac{2r_1}{1+6r_1}(1-6r_1)C & \frac{1+4r_1^2C^2}{1+6r_1} \end{bmatrix} \begin{bmatrix} 1-6r_2 & 2r_2 C \\ \frac{2r_2}{1+6r_2}(1-6r_2)C & \frac{1+4r_2^2C^2}{2+6r_2} \end{bmatrix} = \begin{bmatrix} a_3 & d_3 \\ C_3 & b_3 \end{bmatrix} \tag{7}$$

where

$$C = \cos\sigma_1 h + \cos\sigma_2 h + \cos\sigma_3 h,$$

$$a_2 = (1-6r_1)(1-6r_2) + \frac{4r_1 r_2 C^2}{1+6r_2}(1-6r_2) \quad ,$$

$$b_2 = \frac{4r_1 r_2 C^2(1-6r_1)}{1+6r_1} + \frac{2r_1 C(1+4r_1^2C^2)(1+4r_2^2C^2)}{(1+6r_1)(1+6r_2)},$$

$$C_2 = \frac{2r_1 C(1-6r_1)(1-6r_2)}{1+6r_1} + \frac{2r_1 C(1-6r_2)(1+4r_2^1C^2)}{(1+6r_1)(1+6r_2)},$$

$$d_2 = 2r_1 C(1-6r_1) + 2r_1 C\frac{1+4r_2^2C^2}{1+6r_2}$$

The characteristic equation of the matrix (7) is

$$\lambda^2 - (a^2+b^2)\lambda + a_2 b_2 - c_2 d_2 = 0 \tag{8}$$

The sufficient and necessary conditions for the roots of quadraic equation (8) to be $|\lambda| \leqslant 1$ are

$$a_2 + b_2 | \leqslant 1 + a_2 b_2 - c_2 d_2 \leqslant 2 \tag{9}$$

By considering the inequality of (9), the areas which satisfy Von–Neumenn conditions is (6).

In the Folling we will verify that (6) is the sufficient stability condition of scheme (4).

For eigenvalue λ_i of the matrix G, there is a corresponding unit eigenvector e_1. Then from e_1, we can get an orthonomal basis e_1,e_2 in C^2 space. Set $U = (e_1, e_2)$, it is easy to know that U is a unitary.

Since eigenvalues of similar matrixes are same, we have

$$G^n = U\begin{bmatrix} \lambda_1^n & e_1^H G e_2(\lambda_1^{n-1} + \lambda_1^{n-2}\lambda_2 + \cdots + \lambda_1\lambda_2^{n-2} + \lambda_2^{n-1}) \\ 0 & \lambda_2^n \end{bmatrix}U^H$$

$\| G \|_2$ is a uniformly bounded nuber when R_1 and R_2 satisfy the condition (6). We denote this bounded number by M_o.

Since $M_1 = |\lambda_1||\lambda_2| = \left|\frac{(1-6r_1)(1-6r_2)}{(1+6r_1)(1+6r_2)}\right| < 1$, we have min $(\lambda_1|, \lambda_2|) \leqslant \sqrt{m_1} < 1$.

For any arbitrary σ. There is a eigenvalue whose absolute value is less or equal to the other eigenvalue on absolute value, we might as well suppose that $\zeta_1| \leqslant \xi_2|$ From above discussion, we get

$$\| G^n \|_2 \leqslant 2 + \frac{M_o}{1-M_1}$$

$\| G^n \|_2$ is a uniformly bounded nuber on σ and n.

The whole theorem is proved.

Since the stability condition of the implicit–explicit scheme with explicit form is $r < 1/3$, we can know that it is stable only if $r_1 + r_2 < 2/3$ when a uniform time step is used. Above discussion shows if R_1 and R_2(correspondent r_1 and r_2) are chosen to satisfy condition (6), we will get a stable difference scheme. From (6), we can see $r_1 + r_2 < \frac{2}{3}(2+\sqrt{5})$ in general.

III. Numerical Example

Example: Consider the homogeneous problem of initial and boundary value problem (1).

$$\varphi(x,y,z) = 640xyz(1-x)(1-y)(1-z),$$

a space step $h = 1/20$, mesh ratios are $r_1 = 2.53$ and $r_2 = 0.178$, time step $\tau_1 = 2.53h^2/a$ and $\tau_2 0.178h^2/a$ (a is a positive constant number).

We use difference scheme (4) to calculate approximate solutions. We use the mesh ratio r_1 when k is a odd number and we use the mesh ratio r_2 when k is an even number.

Table 1 Values of Difference Solution $u_{i,j,10}^8$

i \ j	2	4	6	8	10
2	0.415	0.841	1.167	1.365	1.435
4	0.841	1.577	2.189	2.561	2.694
6	1.167	2.189	3.304	3.551	3.734
8	1.365	2.561	3.551	4.156	4.270
10	1.435	2.694	3.734	4.370	4.586

Because of the symmetry of the initial value and calculating scheme in this example, the difference solution is also symmetry in x, y and z directions. The table 1 only shows a quarter of the numerical values in $z = 1/2$ plane. From the table 1, we can see that the difference solutions are convergent.

In the following we use the alternate explicit scheme (2) to calculate the same example. We still choose space step h1/20 and a mesh ratio be the uniformly largest stable ratio $r = 1/3$. So the time step is $\tau = h^2/(3a)$. The computational results are show on table 2.

Table 2 Values of Difference solution $u_{i,j,10}^{32}$

i \ j	2	4	6	8	10
2	0.487	0.922	1.262	1.476	1.549
4	0.922	1.745	2.388	2.795	2.933
6	1.262	2.388	3.268	3.824	4.104
8	1.476	2.795	3.824	4.474	4.696
10	1.549	2.933	4.104	4.696	4.929

By the comparison of Table 1 and table 2, we know, the difference solutions of the eighth level in time direction which are solved by the method proposed in this paper have faster convergence than the difference solutions of the thirty–second level in time direction which are solved by the alternate explict scheme (2), i.e., the time step of the mew-method can be chosen four times more than the largest stability time step of the old method. So we only spent one–fourth the computational amount as before to get the same result in the situation of no raising computational complexity.

Remark. The method proposed in this paper can be used to any explicit scheme to improve stability conditions.

CFD FOR AIR INTAKE INTEGRATION

V. Maudet*, P.Perrier**

Dassault-Aviation, B.P. 300,92211 Saint Cloud, France

* Engineer; ** Head of Theoretical Aerodynamics

Abstract: To design an hypersonic aircraft by experimental way only, would be unacceptable because of the difficulties and so of the induced cost. We have to rely on a mix of theoretical (CFD) and experimental approaches. The methodology of design with CFD rely on check-points based on levels of software and on levels of validation of the codes used in design. major targets of preliminary CFD effort is to demonstrate to the customer that one has confidence in computertools and, so, in the results of computation. Actually CFD is the only way to build complete engine integration, to share responsabilities between cooperants and to clarify interface problems. On air intakes the effort has to cover performance evaluation: prediction of flow field quality (steady and unsteady), and effects of start and unstart of supercritical flows.

Introduction

The higher the Mach number is, the higher will be the necessity of integration of the shapes due to the smaller area between the bow shock-wave and the vehicle. The use of CFD may afford a decisive progress if its analysis capabilities are sufficient. It is only with such a CFD integration that hypersonic air intakes can be efficiently designed[1].

On one hand we will present CFD requirements[1] to better understand the air intake/air frame interactions, on the other hand we will propose a methodology of air intake integration by CFD, supporting by an example.

CFD Requirements and checking

A complete capability is required for global evaluation of performance and for analysis. Such an analysis is better done with reduced CFD effort on not too complex configuration and on elementary box devoted to some phenomena that have to be mastered. But for global evaluation, we need the complete, hence highly complex geometry and the ability to cope with the more complex physical modelling (turbulence + real gases)[3].

The requirements can be summarized in the following table :

Table I Requirements for CFD[4]

A - Physical modelling level required

- real gases + chemistry
- laminar-transitional-turbulent flows

B - Basic phenomena to be included with accuracy

- shock-boundary layer interaction problems
- shock-shock crossing and shock impingement
- turbulent viscous interactions problems
- unsteadiness of flow with separation
- reattaching flows
- vortical flows

C - Basic geometry/boundary conditions to be handled

- rounded nose, complex topology of fuselage, wing, air intakes
- lips and boundary layer diverters, corner flows
- airflows at entrance of engines and/or thermal blocage
- bleed and succion devices
- variable geometry devices.

Concerning the complex configurations, we have to check a minimum number of mesh points, required for the correct rebuilding of the flow-field.

One can validate the code by using it on elementary problems, as it is used in industrial application- or by using it on generic shapes that do not include the maximum complexity of real geometry, but include all the ingredients of the complexity of the flow one by one, at least in one of the subdomains.

For the global validation, the direct analysis of the flow field has to check that all the features of the flows are present, as it is in experimental tests and that the induced level of pressure or heat fluxes is inside experimental scatter of data.

The requirement for efficient CFD code has to cover the items listed in following table :

Table 2 Requirements for code assessment

- multidomain approach, including analytical regular domains compatible with validation of fluid problems.
- multigrid and ability to check convergence with mesh size
- automatic mesh refinement for complex flow and geometry
- alternative physical modelling (physical properties as state equation, viscosity ...)
- unsteady Navier-Stokes with alternative average turbulence modelling when needed but able to return to Prandtl assumption boundary layers Euler flow on subdomains as numerous as possible.
- interface with geometrical definition to check the accuracy of geometrical modelling
- portability on various computers

Methodology of air intake integration

Taking into account the flight envelope in Mach number, angle of attack and sideslip, it is clear that massflow adjustement, to avoid any external perturbation becomes an untractable task. The major part of CFD effort will be devoted to out of adaptation computations.

However the basic adaptation remains a design way to reduce the bad interactions. So the current methodology will go from fitting the geometrical shapes to some design points to making an evaluation of an operation out of adaptation.

An efficient way of making analysis of the forces building is to deduce from the area distribution along X-axis and from the drag distribution an equivalent Cp ratio (the drag variation to the section variation ratio). The integral of this curve (CpdA) gives the evaluation of the thrust or drag of a given configuration, if we know the pressure everywhere thanks to CFD. It is convenient to part external and internal flows in such an analysis. External flows give contribution to drag including lift-induced drag; internal flows must follow Cp versus A (Area) curves not far from the monodimensional isentropic or polytropic flows in air intake. Three dimensional losses appear as a loss against optimum monodimensionnal flows.

So we get an exact evaluation of all the contributions to the propulsive loop if we get the pressure everywhere on the complete aircraft. Such a set of data is clearly the necessary base for final integration evaluation. It comes directly from CFD capability on complex configurations to rebuild the complete pressure distribution including the lips with the boundary layer region where measurements are scanty.

Example

As an example of such an approach, some results obtained from EULER and boundary layer computations, on Star[6] vehicule (hypersonic aircraft) are presented on the following figures. Fig.1 presents the initial mesh before refinements. On Fig.2 is presented a cut into the symmetry plane, showing the different shocks. From the pressure distribution given by computations, we deduce the evolution of the equivalent Cp versus the area for the internal and external flow (see Fig.3). These curves, which the integral gives the total drag or thrust, show how the air intake integration has to be done with the best accuracy.

Concluding remarks

Integration of air intake by CFD is the only way to improve thrust minus drag and selecting of viable candidates to detailed experimental analysis in wind tunnel and after that in flight. From now an efficient methodology seems possible if a CFD capability of taking into account accurate complete geometry is available. It has been fixed that at present state of the art, progresses have to be done on two directions :

-physical modelling improvement by carefull evaluation of experimental tests, done or to be done. For advanced workshops, a main emphasis is on carefull experimental test with better and more complete set of data.

- first reference tests to improve or extend CFD[7] capabilities to identify unsteadiness or transitional flows in air intake, in order to predict the boundaries of correct operation and the margins associated with the catastrophic degradation of the flow quality. Furthermore the flow unsteadiness has to be clearly evaluated at the edge of present CFD. However transition phasis in the flow building around the configuration can be obtained by CFD and its validation is a major challenge at the present time. Unsteadiness evaluation and validation remain a target for the near future[5]

Références

(1) P.Perrier "Concepts of Hypersonic Aircraft", Third Joint Europe/US, Short Course ın Hypersonıc RTW Aachen FRG 1990

(2) V.Maudet, P.Perrier "Air Intake Integration by CFD at High Mach Number" AIAA Second International Aerospace Plane Conference, Paper n° 90-5205, 29-31 October, 1990, Orlando, FL.

(3) M.Mallet, J.Periaux, P.Perrier, B.Stoufflet,"Flow Modelization and Computational Methodologies for the Aerothermal Design of Hypersonic Vehicles: Application to the European Hermes". AIAA Thermophysics, Plasma dynamics and lasers conference, Paper n°88-2628, San Antonio, Texas, June 27-29, 1988.

(4) M.Mallet, J.Periaux and G.Roge, "development of Finite Element Methods for Compressible Navier Stokes Flow Simulations in Aerospace Design". AIAA Aerospace Sciences Meeting, Paper n° 90-0403, Reno, Nevada, January 8-11, 1990.

(5) David H. Campbell,"F-12 series aircraft propulsion system performance and development", AIAA 5th Aircraft Design, Flight Test and Operations Meeting, Paper n°73-821, St. Louis, Missouri, August 6-8, 1973.

(6) M.Rigault,"A Methodology for the Concept Definition of Advanced Space Transportation Systems", conference EAC 89, Bonn-Bad Godesderg, 23 may 1989.

(7) Wolfgang Schmidt "Aerodynamics of High Speed Air Intakes", Status Report on FDP - WG13 Agard Pep 75th Symposium Madrid - May 1990

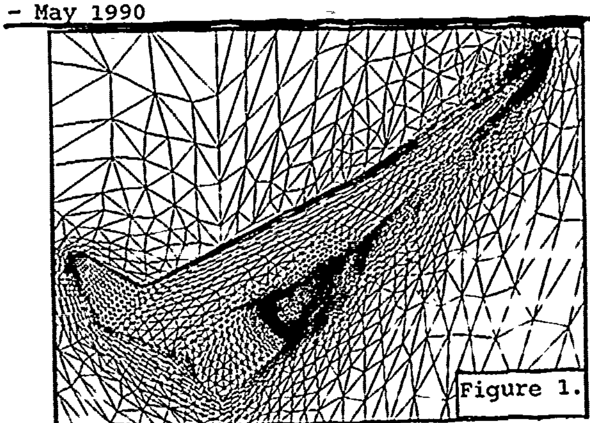


Figure 1.

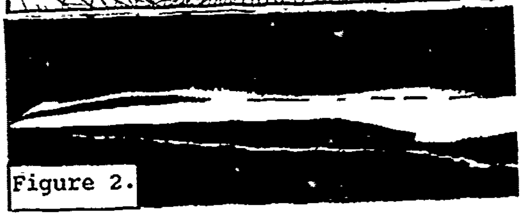


Figure 2.

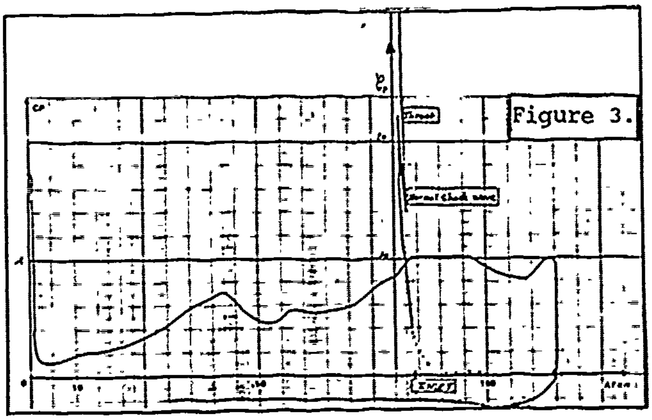


Figure 3.

Discrete models for the analysis of $2D$ wakes in unsteady aerodynamics

G. Riccardi, A. Iafrati and R. Piva
Dip. Meccanica e Aeronautica
Universitá di Roma "*La Sapienza*"
Via Eudossiana, 18 Rome ITALY

Abstract A comparative numerical study of two-dimensional wake dynamics is presented. An application to the free wake motion behind an elliptical loaded wing on the Trefftz plane is considered first. The wake motion is approximated either by the dynamics of a set of vortices (vortex method) or by the dynamics of a piecewise linear curve (boundary element technique). In both approaches some considerations about the time integration accuracy control are made in terms of the most sensitive flow first integral: the Hamiltonian. Two types of the vorticity generation mechanism are tested in the second part of the paper where the application to the flow around a lentil at large incidence in an uniform stream is analyzed. Some aspects of the coupling between the vorticity production and the wake interaction are discussed in order to explain the periodic vorticity shedding.

1 Introduction

In the present paper we perform a comparative analysis between two different discretization techniques for the wake dynamics. the *vortex method* and the *boundary element* method. In particular we consider a free wake as well as the two wakes generated by a lentil in an uniform stream at large incidence. The latter case gives the opportunity to investigate the interaction mechanism between the wakes.

As a preliminar case we take into consideration the dynamics of a free wake. In particular we develop a numerical study for the motion of a vortex sheet behind an elliptically loaded wing on the Trefftz plane. A large emphasis is placed on the time integration accuracy control.

A study of the wake motion behind a lentil at large incidence is carried out in the second part of the paper. This case is complicated by the coupling with the vorticity production whose model is, in turn, strongly dependent on the wake dynamics. Typically the interaction between the upper and lower wakes leads to a periodic vorticity shedding. We have adopted two discrete approaches in order to study the kinematic of the flow field about the body. i.e. *conformal mapping* and *boundary elements*.

2 Free wake dynamics

The vorticity field of a $2D$ wake is given by

$$\omega(x,t)=\int_0^{L_t}\gamma_t(\sigma_t)\delta\left(x-x_c(t,\sigma_t)\right)d\sigma_t$$

in which γ_t is the derivative with respect to the current natural parameter s_t of the circulation $\Gamma_t(s_t)$ on each circuit around the wake arc $C_t\,|_{[0,s_t]}$ and δ is the Dirac measure on $\mathcal{R}^2$. This vorticity field induces the velocity

$$\forall x\notin C_t \;\; : \;\; u(x,t)=\int_0^{L_t}\gamma_t(\sigma_t)K\left(x-x_c(t,\sigma_t)\right)d\sigma_t$$

where $K=\nabla^{\perp}G$ is the *Biot-Savart kernel* and $G(x)=\dfrac{1}{2\pi}log|x|$. This field has a jump across the wake given by

$$[u]=\gamma_t\tau \quad ,$$

and the use of the tangential wake velocity definition

$$w_\tau=\frac{1}{2}(u_\tau^+ + u_\tau^-)$$

leads to the classical Birkhoff initial value problem for the wake motion [4]. Its numerical integration is carried out employing a preliminary kernel treatment that, avoiding the computation of a Cauchy integral, simplifies the overall discrete representation of the wake. If K^* is the new kernel we consider the "treated" initial value problem

$$\left\{\begin{array}{lcl}\dfrac{\partial x_l^*(t,s_0)}{\partial t} & = & \displaystyle\int_0^{L_0}K^*(x_l^*(t,s_0)-x_l^*(t,\sigma_0))\gamma_0(\sigma_0)d\sigma_0 \\ x_l^*(0,s_0) & = & x^0(s_0) \quad .\end{array}\right. \tag{1}$$

Following the classical vortex method approach we make here a convolution of the physical kernel with a suitable cutoff function in order to remove its non-integrable singularity in zero. We consider two forms of treatment which lead to the desingularized kernel

$$\epsilon\in\mathcal{R}^+ \;,\; K_\epsilon^{des}(x)=\frac{1}{2\pi}\frac{x^{\perp}}{|x|^2+\epsilon^2}=K(x)\left(1-\frac{1}{1+\left(\frac{|x|}{\epsilon}\right)^2}\right) \quad ,$$

and to the regularized (by gaussian functions) kernel [1]

$$\delta\in\mathcal{R}^+ \;,\; K_\delta^{reg,n}(x)=K(x)\left(1-\sum_{k=1}^{n}b_k e^{-c_k\frac{|x|^2}{\delta^2}}\right) \text{ with } \sum_{k=1}^{n}b_k=1 \quad .$$

In the following we indicate by K_ρ^* both the desingularized and the regularized kernel, where ρ identifies with ϵ and δ respectively.

The wake is approximated with a set of vortices (Vorticity Blobs method) or with a straight panel representation (Boundary Elements technique). In the former case we study the vortex dynamics by performing an approximate integration of the following ordinary differential system of N_{VB} equations

$$\left\{\begin{array}{lcl}\dfrac{dx_i^{VB}}{dt} & = & \displaystyle\sum_{j=1}^{N_{VB}}\Gamma^j K_\rho^*(x_i^{VB}-x_j^{VB}) \\ x_i^{VB}(0) & = & x^0(s_0^i)\end{array}\right.$$

where Γ^j is the circulation around the $j-th$ vortex. In the second approach we study the evolution of the piecewise linear curve approximanting the wake by following the motion of its vertices. This leads to the system of N_{BE} equations

$$\begin{cases} \dfrac{dx_i^{BE}}{dt} & = \displaystyle\sum_{j=1}^{N_{BE}} \gamma_t^j \int_{s_t^j}^{s_t^{j+1}} K_\rho^*(x_i^{BE} - x^{BE}(t,\sigma_t))d\sigma_t \\ x_i^{BE}(0) & = x^0(s_0^i) \end{cases}$$

where γ_t is assumed constant along any panel.

A direct evaluation of the numerical time-integration accuracy in both VB and BE discrete approaches is possible computing suitable quantities related at some flow first integrals. From our numerical experience we deduce that the most sensitive flow invariant is the *Hamiltonian* defined by

$$\mathcal{H}_\rho^* \equiv \int_0^{L_t} d\sigma' \gamma_t(\sigma') \int_0^{L_t} d\sigma'' \gamma_t(\sigma'') G_\rho^* (x_c^*(t,\sigma') - x_c^*(t,\sigma'')) \quad (2)$$

in which for the two above choices of K_ρ^* the smoothed Green's functions are

$$G_\epsilon^{des}(x) = G(x) + \frac{1}{4\pi} log\left(\frac{|x|^2 + \epsilon^2}{|x|^2}\right)$$

$$G_\delta^{reg,n}(x) = G(x) + \frac{1}{4\pi}\sum_{k=1}^{n} a_k Ei(c_k \frac{|x|^2}{\delta^2}) \quad .$$

It worths to point out that $\mathcal{H}_\rho^{VB,*}$ is also a first integral for the discrete vortex model whereas $\mathcal{H}_\rho^{BE,*}$ is not a stationary quantity. In fact the flow maps a set of point in a set of points but it does not transform a set of straight segments in a set of straight segments. Hence

$$\frac{d\mathcal{H}_\rho^{BE,*}}{dt} = 2\sum_{k=1}^{N_{BE}} \gamma_0^k \int_{s_0^k}^{s_0^{k+1}} d\sigma_0 u_\rho^\perp(x^{BE,*}(t,\sigma_0),t) \cdot \frac{dx^{BE,*}(t,\sigma_0)}{dt} \neq 0$$

where u_ρ is the approximate self-induced velocity field, while $\dfrac{dx^{BE,*}}{dt}$ is its linear interpolation given by the BE description of the wake motion. The numerical integration of the above equation allows for a control of consistency between the wake velocity field and its displacement by comparing with the numerical values obtained by the corresponding discretized form of (2).

We apply the previous techniques to the numerical simulation of a wake shedded from an elliptically loaded wing on the *Trefftz plane* (see *fig.*1). As already described in [4] (where a VB approach with a desingularization of the Biot-Savart kernel was used), both discrete approaches for a fixed smoothing parameter ρ lead to a convergent numerical solution of equation (1) with respect to the refinement of both the initial space discretization and the time integration step. Then it is possible to find *for any* ρ a well defined limit solution, but unfortunately the resulting sequence for $\rho \rightarrow 0^+$ does not converge. However, as shown in [4], the loss of convergence appears confined in a smaller and smaller neighbourhood of the sheet tips. These calculations show that the time integration accuracy control requires a very large computational effort for the BE approach, whereas it results simple and fast for the VB formulation In fact the VB approach appears faster than the BE formulation even if the time integration accuracy control is neglected.

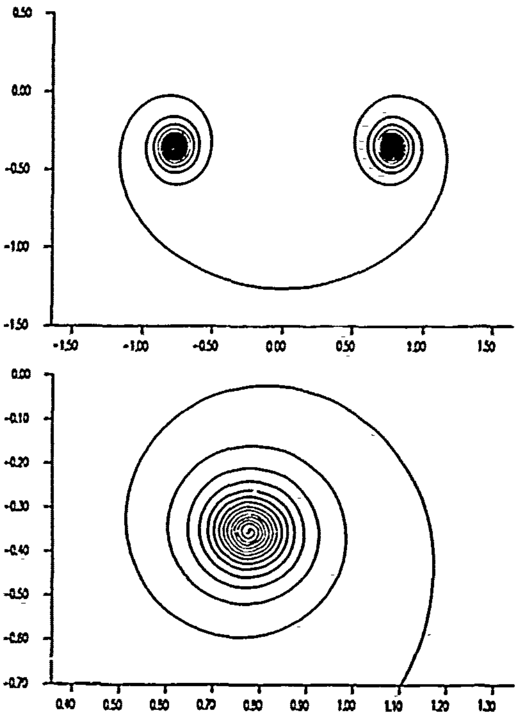


Fig.1 *Wake configuration as interpolating curve of the vortex-blob positions* ($\epsilon = 0.05$)

3 Wake dynamics past a lentil

The above numerical techniques are applied to the simulation of the flow around a lentil placed in an uniform stream at large incidence. To this aim we have to introduce models for the velocity field about the body and for the vorticity production at the sharp edges where the separation occurs.

Velocity field representation

The first task is reached by following two different paths. the conformal mapping analysis [2] and the Poincaré representation [3].

In the former way the field external to the body (z *plane*) is mapped in the exterior of a unit circle (ς *plane*) by the Karman-Trefftz transformation. The wakes are discretized with a suitable set of vortices (to be considered points out of the flow field) and the motion of the k th vortex is obtained by a numerical integration of the equation

$$\frac{dz_k}{dt} = \frac{d}{d\varsigma}\left(W - \frac{\Gamma_k}{2\pi i}\frac{1}{\varsigma - \varsigma_k}\right)\frac{d\varsigma}{dz}\bigg|_{\varsigma_k} + \frac{1}{2}\frac{\Gamma_k}{2\pi i}\frac{d}{d\varsigma}\left(\frac{d\varsigma}{dz}\right)\bigg|_{\varsigma_k}$$

together with the smoothing procedure [5] for the vortex interactions.

Using the Poincaré representation we decompose the velocity field in the sum of a uniform translational flow u_∞ plus a pertubation field u generated by the presence of the body. The limit of this field on the body is obtained by solving the integral equation

$$\frac{1}{2}u_\tau(\xi) - \int_{\partial\Omega_B} u_\tau(\eta)K(\xi-\eta)\cdot\tau(\xi)ds(\eta) =$$
$$\int_{\partial\Omega_B} u_\infty\cdot\nu(\eta)K(\xi-\eta)\cdot\nu(\xi)ds(\eta)+$$
$$\sum_{n=1}^{2}\tau(\xi)\cdot\int_{\mathcal{W}_n(t)}\gamma_t^n(\sigma_t)K\left(\xi-x_n(t,\sigma_t)\right)d\sigma_t$$

where (τ,ν) is the intrinsic reference system on the body boundary $(\partial\Omega_B)$ and $\mathcal{W}_{1,2}(t)$ is the configuration of the wake 1 and 2 respectively. In the computational procedure the two issuing wakes may be discretized either by VB or by BE technique.

Production of vorticity

When a vortex discretization of the wake is adopted, we consider two different schemes for the production of vorticity. In the first one (used only in the conformal approach) the nascent vortices are placed on the symmetry plane of the body at a small distance from the corresponding edge and their intensity is fixed by imposing *a finite velocity* at the lentil edges. In the second one (adopted also in the Poincaré formulation) we follow the motion of *a neutral particle* which starts from the body edges with velocity

$$w = \frac{1}{2}\left(u(P^+) + u(P^-)\right)$$

(where $P^{+,-}$ are points near the edge on opposite sides of the wake) during a suitable time interval. Hence we assign a circulation around the particle obtained by integrating the vorticity generated in that time interval. A similar approach is adopted also with a panel discretization of the wake, but the amount of shedded vorticity is scattered along the nascent panel.

In the present numerical simulations we assume the incidence of the lentil sufficiently large to have two separation points located on the body edges. Two wakes are produced behind the body and their interaction plays an essential role in the flow field development: the main effect is a periodic wake detachment that leads to the formation of the typical vortex street shown in *fig.2*.

The comparison between the conformal mapping and the Poincaré description of the velocity field gives a good agreement with respect to certain global flow quantities such as the body circulation Γ or the time history of $\frac{d\Gamma}{dt}$, even by assuming different vorticity production mechanisms. From this comparative analysis the method of the *neutral particles* appears more sensitive to the velocity perturbations near the edges than the *finite velocity* approach. Moreover the comparison between the two wake disretizations has shown an equal mean behaviour with some superposed oscillations that appear more persistent for the BE technique, owing to the wake continuity constraint.

The flow resulting from an impulsive starting evolves towards a periodical one after a short initial transient. This periodicity is forced by the process of interaction between the wakes, shown in *fig.3*, that can be summarized as follows. Alternatively one wake (say 1) rolls up, and leaves the generating edge inducing the convection of the other wake (say 2) into the region between the first one and the body. The motion of the wake 2 leads to a strong *local* stretching of the wake 1 up to its breaking in a region close to the edge. When the velocity field induced by the lower wake (see *fig.3b*) is sufficiently high to detach one or more elements from the upper wake the above phenomenon is accompanied by the convection of these elements up to the lower edge. This kind of interaction leads to the formation of several small clusters in which the elements coming from the upper wake couple with some elements of the lower wake. The presence of these clusters in the neighbourhood of the edge influences the local velocity field and delays the new formation of the lower wake. Moreover their effect on the vorticity production mechanism enhances the oscillatory behaviour of $\frac{d\Gamma}{dt}$, even if a separation of this effect from the self-induced one is difficult. We observe this mechanism only for the lower edge, while it may occur also at the upper edge only for large incidence of the lentil.

In order to reduce the CPU time we adopt a coalescence procedure of the far wake regions replacing a cluster of wake elements with a single vortex. At the present time the effects of this approximation are not completely understood: it seems that this causes a little reduction of the circulation relased and a short delay in the time history of the circulation around the body. Also the choice of the coalescence parameters (and obviously the choice of the procedure itself) is very critic because we have experimented that this can influence the vorticity production in a very subtle way.

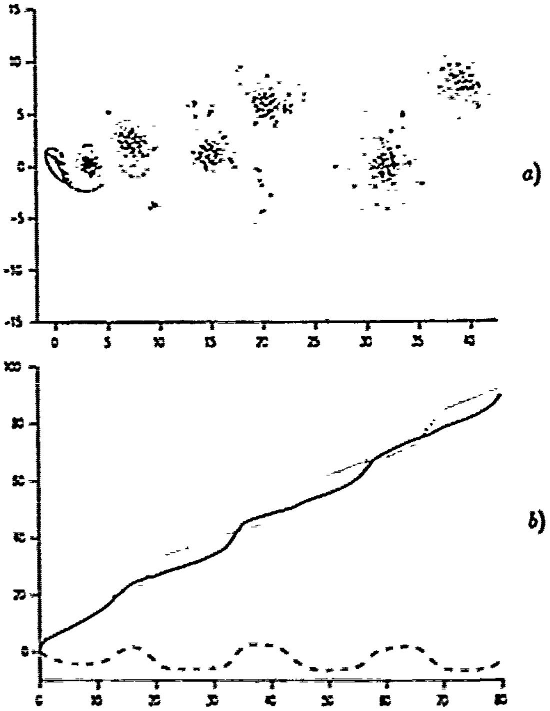


Fig.2 *a) Vortex street behind a lentil at incidence* 60°. *b) Circulation shedded at the leading-edge* (dotted line), *trailing-edge* (solid line) *and global circulation around the wakes* (dash-dot line)

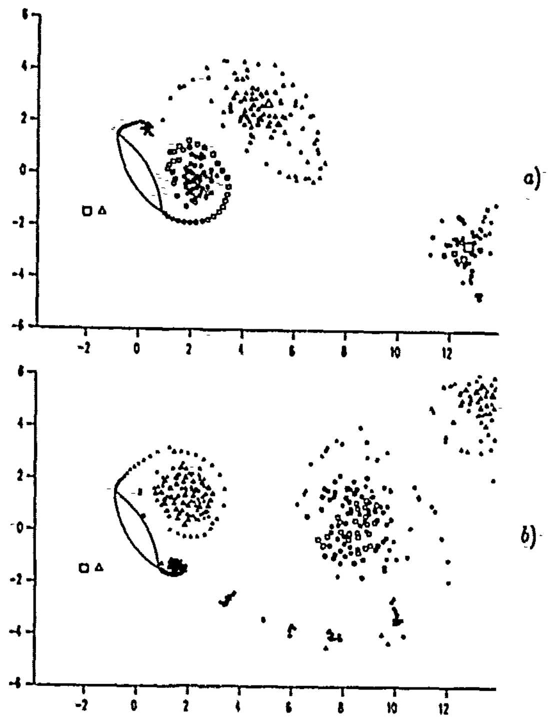


Fig.3 *Flow field past a lentil: detachment of the upper (a) and lower wake (b)*

Finally carrying out several other calculations with different incidences it appears a monotonic decrease of the vorticity shedding frequency for growing incidence which obviously vanishes when the lentil is normal to u_∞. We have also analyzed the flow field for different values of the lentil thickness at a fixed incidence of 60°: the shedding frequency decreases monotonically with the lentil thickness up to reaching the flat plate frequency. This behaviour appears essentially due to the increasing distance between the wakes that slows down their interaction.

References

[1] Beale, J.T. and Majda, A. *High order accurate Vortex Methods with explicit velocity kernels*, J. Comput. Phys. 58 (1985), pp. 188-208

[2] Iafrati, A., Riccardi, G. and Piva, R. *Interazione di scie a valle di una lente*, XI Congresso AIDAA Forlì 1991

[3] Bassanini, P., Casciola, C.M., Lancia, M.R. and Piva, R. *A boundary integral formu, for the kinetic field in aerodynamics*, European J. Mech. B/. uids xx (1990), pp.

[4] Krasny, R. *Computation of vortex sheet roll-up in the Trefftz plane*, J. Fluid Mech. 184 (1987), pp. 123-155

[5] Kiya, M. and Arie, M. *A contribution to an inviscid vortex-shedding model for an inclined flat plate in uniform flow*, J. Fluid Mech. 82 (1977), pp. 223-240

INITIAL TEMPERATURE FIELD FOR UNSTEADY LAMINAR FORCED CONVECTION FROM AN IMPULSIVELY STARTED SPHERE

Lai-Chen Chien
Institute of Physics, Academia Sinica
TAipei, Taiwan 11529
Republic of China

Abstract-Analytic solution for forced convection heat transfer from an impulsively started heated sphere is investigated. Because of the impulsive start, there is a singularity at the very beginning of the motion. The accurate analytic solution for the initial temperature field is obtained by solving the non-linear energy equation using the method of matched asymptotic expansion to the third order. The solution is in terms of exponential function and error function. The time development of the temperature field is plotted and investigated. The local Nusselt number over the sphere surface and the progress of minimum Nusselt number point with time are obtained.

I. INTRODUCTION

The problem of incompressible viscous flow over an impulsively started sphere has been studied by many investigators (Bentwich and Miloh, 1978). The extension of the problem to the heat transfer has also been the popular subject of numerous analytical and experimental investigations (Chen and Mucoglu, 1977). Several theoretical investigations have been reported centering around the classical problems of heat and mass transfer from a solid sphere into a low Reynolds number velocity field. Sano (1981), Bentwich and Miloh (1978) obtained the asymptotic solutions for Stokes-Oseen flow using the method of matched asymptotic expansions. Acrivos and Taylor (1962) used singular perturbation technique expressed the Nusselt number in terms of Peclet number, and yielded an accurate expression for the rate of transfer of energy for small Reynolds number. Hieber and Gebhart (1969) studied mixed convection from a heated sphere for Stokes flow. Chen and Mucoglu (1977) solved the conservation equations of the non-similar boundary layer using finite difference method to study the combination forced and free convection about a sphere for small Reynolds number.

In this study, we try to employ the inner-outer expansion method to solve the axisymmetrical unsteady Navier-Stokes and energy equations in an attempt to extend the boundary layer theory to larger Reynolds number. The stream function for the initial flow obtained by the matched asymptotic expansion to the third order (Chien and Chen, 1984) is used to solve the energy equation. The analytic solution to the third order for the temperature distribution corresponding to the initial flow field is treated with great care because of the complicated mathematical operation. The temperature field is thus obtained in terms of exponential and error function. The time development of the temperature field properties such as the local Nusselt number over the sphere surface, and the progress of minimum Nusselt number point with time is investigated.

II. BASIC EQUATIONS

Consider a solid sphere of radius r_0 which is started impulsively from rest and subsequently moves with constant velocity U. The origin of the spherical coordinate system is fixed at the center of the sphere with the axis $\theta=0$ in the direction opposite to the motion of the sphere. Assume that the fluid is an incompressible viscous continuum with constant properties, and that the effects of the heating of the fluid by viscous dissipation from sphere surface is neglected. With the foregoing assumption, the Navier-Stokes equation for the fluid motion can be expressed in the form (Chien and Chen, 1984) as

$$\left[\frac{\partial}{\partial t}-\frac{\varepsilon}{r^2\sin\theta}\left(\frac{\partial\psi}{\partial\theta}\frac{\partial}{\partial r}-\frac{\partial\psi}{\partial r}\frac{\partial}{\partial\theta}+2\cot\theta\frac{\partial\psi}{\partial r}-\frac{2}{r}\frac{\partial\psi}{\partial\theta}\right)\right]\nabla^2\psi$$

$$=\alpha\varepsilon^2\nabla^4\psi \qquad (1)$$

where t is non-dimensionalized by the characteristic time t_0, radial coordinate r by the radius of the sphere r_0, stream function by r_0U. The parameters are defined by $\varepsilon=Ut_0/r_0$, $\alpha=1/(\varepsilon Re)$, where Reynolds number is based on radius. For the problem of initial flow, ε is a small eqantity much less than 1 and α is order 1. The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.

The flow field is divided into two regions. One is the inner region close to the sphere surface and the outer is the outer inviscid region, i.e. $\psi=\psi^0+\psi^1$. The outer solution to the third order for the stream function is obtained by Wang (1969). The inner solution to the third order by the method of inner-outer expansion is given by Chien and Chen (1984). The composition stream function is

$$\psi=\frac{1}{2}\left(r^2-\frac{1}{r}\right)\sin^2\theta+3\frac{\tau}{\sqrt{Re}}\left[-\frac{1}{r\sqrt{\pi}}+\frac{3\tau}{r^2\sqrt{\pi}}\left(\sqrt{2}-1+\frac{2}{9\pi}\right)\cos\theta\right.$$

$$-\frac{1}{2r}\frac{\tau}{\sqrt{Re}}+\left(\frac{1}{\sqrt{\pi}}e^{-\eta^2}-\eta\,\mathrm{erfc}\,\eta\right)+\frac{\tau}{\sqrt{Re}}\left[\left(\eta^2+\frac{1}{2}\right)\mathrm{erfc}\,\eta-\frac{1}{\sqrt{\pi}}e^{-\eta^2}\right]$$

$$\left.+\frac{3}{2}S(\eta)\cos\theta\right]\sin^2\theta+2\left(\frac{\tau}{Re}\right)^{3/2}$$

$$\left[\int_0^\eta f_1(\eta)d\eta-\left(2\eta^3-\frac{5}{\sqrt{\pi}}\eta^2+\frac{3}{2}\eta\right)\right]\sin^2\theta+18\frac{\tau^2}{Re}$$

$$\left[\int_0^\eta g_3(\eta)d\eta-\frac{2}{\sqrt{\tau}}\left(1-\sqrt{2}\frac{2}{9\pi}\right)\eta\right]\sin^2\theta\cos\theta+27\tau^2\frac{\tau}{\sqrt{Re}}$$

$$\left[2\int_0^\eta f_3(\eta)d\,\sin^2\theta\cos^2+\int_0^\eta F_3(\eta)d\eta\sin^4\theta\right], \qquad (2)$$

where η is defined by $(r-1)/2\sqrt{\alpha t}$, τ is dimensionless time and $S(\eta)$ is

$$S=\frac{3}{\sqrt{\tau}}e^{-\eta^2}\,\mathrm{erfc}\,\eta-\frac{2\sqrt{2}}{\sqrt{\tau}}\mathrm{erfc}\sqrt{2}\eta-\eta\,\mathrm{erfc}^2\eta-\frac{1}{\sqrt{\tau}}\left(\frac{7}{3}+\right.$$

$$-\frac{1}{\sqrt{\pi}}\left(\frac{7}{3}+\frac{4}{9\pi}\right)e^{-\eta^2}+\left(\frac{2}{3\pi}+1\right)\eta\,\mathrm{erf}\,\eta+\frac{2}{3}\left(\frac{2}{3\pi}-1\right)$$

$$(\mu^3 \mathrm{erf}\mu - \frac{1}{\sqrt{\pi}}\mu^2\ e^{-\mu^2}) + \frac{4}{3\sqrt{\pi}}\mathrm{erfc}\mu$$

And f_1, f_3, F_3 and g_3 are polynomials of the similar forms.

In order to solve the energy equation to obtain the temperature field for the convection problem considered, the equation is written as (Chien and Kung, 1982)

$$[\frac{\partial}{\partial t} - \frac{\varepsilon}{r^2\sin\theta}(\frac{\partial\psi}{\partial\theta}\frac{\partial}{\partial r} - \frac{\partial\psi}{\partial r}\frac{\partial}{\partial\theta})]T$$

$$= \frac{\alpha\varepsilon^2}{Pr}(\frac{\partial^2}{\partial r^2} - \frac{\cot\theta}{r^2}\frac{\partial}{\partial\theta} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2})T, \qquad (3)$$

where dimensionless temperature T is $(T' - T_\infty)/(T_W/T_\infty)$, T' is dimensional temperature, and T_W, T_∞ are the temperature on the sphere surface and at infinite respectively. When the sphere is impulsively started to move with a constant velocity, a constant temperature difference between the solid sphere and the fluid is suddenly imposed. The initial and boundary conditions are

$$\begin{array}{llll} & T = 0 & \text{for } t \le 0 & (4a) \\ & T = 1 \text{ at } r = 1 & \text{for } t > 0 & (4b) \\ \text{and} & T \to 0 \text{ at } r \to \infty & \text{for } t > 0 & (4c) \end{array}$$

III. ANALYTIC SOLUTION FOR THE TEMPERATURE FIELD

Similar to the problem of convective heat transfer over an impulsively started circular cylinder (Chien and Kung, 1982), we construct the solution for the temperature field by the method of additive composition. Sano (1978) has shown that the outer solution for the energy equation vanishes concerning the heat transfer for the flow over an impulsively started cylinder. The conclusion is also valid for the sphere problem considered in this investigation.

It can be shown that the inner solution is of the form.

$$T^i = T_1(R,\ \theta,\ t) + \varepsilon T_2(R,\ \theta,\ t) + \varepsilon^2 T_3(R,\ \theta,\ t)$$

where R is the stretched radial coordinate and defined by $R = (R-1)/\varepsilon$. Substituting the above expression into (3) and collecting the terms of the same order ε, we have the equations.

$$T_{1t} - (\alpha/Pr)\ T_{1RR} = 0 \qquad (5)$$

$$T_{2t} - (\alpha/Pr)\ T_{2RR} = (\psi^i_{1\theta}T_{1R} + \psi^i_{1R}T_{1\theta})\sin\theta \qquad (6)$$

$$T_{3t} - (\alpha/Pr)\ T_{3RR} = \alpha(T_{1\theta\theta} - \cot\theta T_{1\theta})/Pr +$$

$$(\psi_{2\theta}T_{1K} - 2K\psi^i_{1R}T_{1K} + \psi^i_{1\theta}\psi_{2R} - \psi^i_{2R}T_{1\theta} +$$

$$2R\psi^i_{1R}T_{1\theta} - \psi^i_{1R}T_{2\theta})/\sin\theta. \qquad (7)$$

for the first, the second and the third order respectively.

And the boundary conditions for the above equations are

$$T\ (R,\theta,t) = 1 \text{ at } R = 0 \qquad (8a)$$

$$T_n(R,\theta,t) = 0 \text{ at } R = 0 \quad \text{for } n = 2,3, \qquad (8b)$$

$$T\ (R,\theta,t) \to 0 \text{ at } R \to \infty \quad \text{for } n = 1,2,3. \qquad (8c)$$

The solution for the first order is similar to that obtained by Sano (1978) solving the convection of cylinder problem,

$$T_1 = \mathrm{erf}\ \mu \qquad (9)$$

where μ is $\sqrt{Pr}\ \eta$ and η is the inner stretched variable $\eta = R/(2\sqrt{\alpha t})$. Substituting the first order solution (9) and the inner stream function into the right hand side of the second order equation (6), one has

$$T_{2t} - \alpha T_{2RR}/Pr = -6\sqrt{Pr}[(\eta - 1\sqrt{\pi})e^{-\eta^2}\mathrm{erfc}\eta$$

$$+ e^{-\xi^2}/\sqrt{\pi}]/\sqrt{\pi} \qquad (10)$$

where $\xi = \sqrt{Pr+1}\mu$. Assume the solution for T be of the form $T_2 = tF_2(\mu)\cos\theta$. Then the equation (10) becomes

$$F_2''(\eta) + 2PrF_2' - 4PrF_2 = 24Pr[(\eta - 1\sqrt{\pi})e^{-\eta^2}$$

$$-\eta e^{-\mu^2}\mathrm{erfc}\eta + e^{-\eta^2}/\sqrt{\pi}]\sqrt{Pr}/\pi. \qquad (11)$$

The homogeneous part of equation (11) is of the form

$$f''(\eta) + 2\eta f' - 2nf = 0. \qquad (12)$$

The solution may be in terms of error function and Hermite polynomial (courant and Hibert, 1953, p.90). The particular integral for (11) is obtained by the method of undetermined coefficients. Then we have the complete solution.

$$F_2(\mu) = [\frac{6(2Pr+1)\sqrt{Pr}}{(3Pr+1)} - 4\sqrt{Pr} - 3C_2\pi/8 - \pi\sqrt{\pi C_2}(3Pr+1)I(\infty)].$$

$$(2\mu^2+1)\mathrm{erf}\mu + 2\sqrt{Pr}/\pi e^{-\mu^2}]/ + Pr(4-3\sqrt{\pi})e^{-\mu^2}/\pi$$

$$+3\sqrt{Pr}/\pi\eta e^{-\mu^2}\mathrm{erfc}\eta - 6\sqrt{Pr}(2Pr+1)/$$

$$[\pi(3Pr+1)e^{-\mu^2+\eta^2} + C_2[-\sqrt{\pi}(3Pr+1)\eta e^{-\mu^2}\mathrm{erfc}\eta$$

$$+3e^{-\mu^2-\eta^2} - \sqrt{\pi}(3Pr+1)(2\mu^2+1)(1(\eta) - 1(\infty))]/8 \qquad (13)$$

where $C_2 = 12(Pr+1)^2Pr/[\pi(3P+1)]$ and $1(\eta) =$ $I(\eta) = \int_0^\eta e^{-\mu^2}\ \mathrm{erfc}\eta d\eta.$

The right hand side of equation (7) is function of derivatives of the first and the second order expansion. Substituting (9) and (13) into the right hand side of equation (7), we have

$$T_{3t} - \alpha T_{3RR}/Pr = [M_{31}(\eta) + M_{32}(\eta)]\cos\theta$$

$$+[M_{33}(\eta) + M_{34}(\eta)]\cos\theta$$

$$+[M_{34}(\eta) + M_{35}(\eta)]\sin\theta \qquad (14)$$

where the coefficients of the triangular functions are polynomial of error and exponential function. We assume the solution of Equation (14) be of the form

$$T_3 = 3\sqrt{\alpha t}Pr/\pi[8F_{31}(\eta) + F_{32}(\eta)]t\cos\theta + [9\sqrt{Pr}/\pi F_{33}(\eta)$$

$$+ 3F_{36}(\eta)]t^2\cos^2\theta + [-9\sqrt{Pr}/\pi F_{34}(\eta)$$

$$+ F_{35}(\eta)/2)]t^2\sin^2\theta. \qquad (15)$$

Substituting the above expression into equation (14), it gives set of six equations in of the form

$$F''(\eta) + 2Pr\eta F' - 2nPrF$$

$$= GH(\mathrm{erfc}\eta,\ \exp(-\eta^2),\ \exp(-\mu^2)) \qquad (16)$$

The completementary solution of (16) is able to be expressed in terms of error function and Hermite polynomial. And the particular integrals are obtained by the method of undetermined coefficient. The complicated mathematical operation is repeated with great care.

To sum up, the solution of the energy equation is

$$\begin{aligned} T = {} & \mathrm{erfc}\mu + \tau F_2(\eta)\cos\theta \\ & + [F_{31}(\eta) + 3\tau\sqrt{\tau}\mathrm{Pr}/(\pi \mathrm{Re})\ F_{32}(\eta)]\cos\theta \\ & + \tau^2[9\sqrt{\mathrm{Pr}}/\pi F_{33}(\eta) + F_{36}(\eta)]\cos^2\theta \\ & + \tau^2[-9\sqrt{\mathrm{Pr}}/\pi F_{34}(\eta) + F_{35}(\eta)/2]\sin^2\theta. \end{aligned} \tag{17}$$

where $F_2(\eta)$, F_{31} to F_{36} are polynomial of error function and exponential function

IV. RESULTS

The temperature fielf (17) is plotted in Figure 1 for Re = 100, Pr = 0.7. At short time after the impulsively start, the isotherms in the front half part are almost parallel to the surface of the sphere. As time goes on, at τ = 0.6, the isotherms displaced away from the sphere as the viscous layer thickens. Similar phenomenon is shown in Figure 2 for Re = 500, Pr = 0.7 at τ = 0.6. the higher the Reynolds number, the thinner the thermal boundary layer because the convection becomes more effective in displacing the streamlines downstream away from the sphere.

Having computed the temperature field, we can estimate the heat transfer between the sphere and the surrounding fluid.

$$\mathrm{Nu} = -K(\partial T/\partial r)_{r=1}.$$

The local Nusselt number distribution around the surface of the sphere for Pr = 0.7 at Re = 100 and 500 is shown in Figure 3. Because no existing investigation is available for comparison, we compare the trends of this solution with those obtained in studying cylinder by Sano (1978), Chien and Kung (1982). The similar results are obtained.

We differentiate the Nusselt number distribution function, set it equal to zero, and find the minimum Nusselt number point. By Newton's method, we can obtain the progress of minimum Nusselt number point at the sphere surface with respect to time, Figure 4.

V. CONCLUSIONS AND RECOMMENDATIONS

The major objective of the present investigation has been to work out the short time solution for unsteady forced convection heat transfer from an impulsively started sphere. Because of the impulsive start, there is a singularity behavior of the flow field at the very beginning of the motion. An accurate solution is obtained by the method of asymptotic expansion to the third order. The expansion is valid for the short time only, $\tau < 1$. The viscous layer considered flow field (Chien and Chen, 1984) and temperature field obtained in this investigation can be used as initial conditions for numerical computation. And the solution can be continued by numerical integration to obtain the larger time solution (Schlichting, 1979, p. 149). Using the accurate stable numerical method to solve the governing equations, we can depict the time history of the flow patterns and the temperature fields.

REFERENCES

1. Acrivos, A. and Taylor, I. D. (1962). Heat and Mass Transfer from Single Sphere in Stokes Flow. Phys. Fluids, Vol. 5, pp. 387-394.
2. Bentwich, M. and Miloh (1978). The Unsteady Matched Stokes-Oseen Solution for the Flov Past a Sphere. J. Fluid Mech., Vol. 98, pp. 17-32.
3. Chen, T. S. and Mutoglu (1977). Analysis of Mixed Forced and Free Convection about a Sphere. Int. J. Heat Mass Transfer, Vol. 20, pp. 867-875.
4. Chien, L. C. and Kung, I. S. (1982). Heat Transfer from an Impulsively Started Circular Cylinder. Computational and Asymptotic Methods for Boundary and Interior Layers, J. J. H. Miller ed., Boole Press, Dublin, Ireland, pp. 177-182.
5. Chien, L. C. and Chen, S. W. (1984). The Initial Flow Past an Impulsively Started Sphere. Computational and Asymptotic Methods for Boundary and Interior Layers, J. J. H. Miller ed., Boole Press, Dublin, Ireland, pp. 179-184.
6. Courant, R. and Hibert, D. (1953). Methods of Mathematical Physics, Vol. 1, Interscience Press, New York.
7. Hieber, C. A. and Gebhart, B. (1969). Mixed Convection from a Sphere at Small Reynolds and Grashof Number. J. Fluid Mech., Vol. 38, pp. 137-159.
8. Gary, J. R. (1953). The Determination of Local Forced Convection Coefficient for Sphere. Trans. Am. Soc. Mech. Eng. Vol. 75, pp. 483-487.
9. Sano, T. (1978). Short-time Solution for Unsteady Forced Convection Heat Transfer from an Impulsively Started Circular Cylinder. Int. J. Heat Mass Transfer, Vol. 21, pp. 1505-1516.
10. Sano, T. (1981). Unsteady Flow Past a Sphere at Low Reynolds Number. J. Fluid Mech., Vol. 112, pp. 433-441.
11. Schlichting, H. (1979). Boundary Layer Theory McGraw-Hill, New York.
12. Vliet, G. C. and Lepper, G. (1961). Forced Convection Heat Transfer from an Isothermal Sphere to Water. J. Heat Transfer, Vol. 83c, pp. 163-175.
13. Wang, C. Y. (1967). The Flow Past a Circular Cylinder Which is Started from Rest. J. Math. Phys., Vol. 46, pp. 195-202.

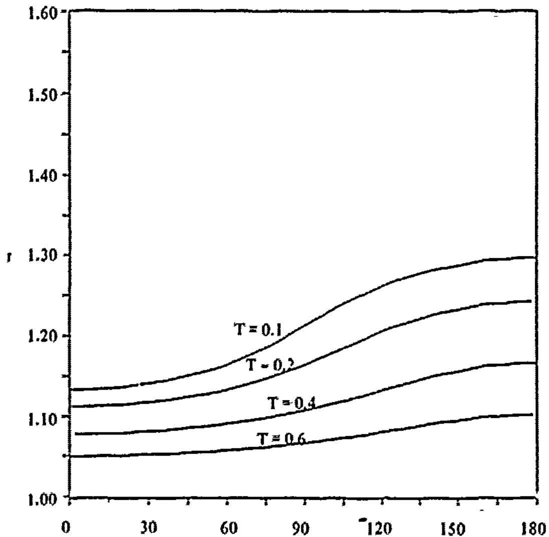


Figure 1. Isothermal for Pr=0.7 at Re=100, Time=0.6.

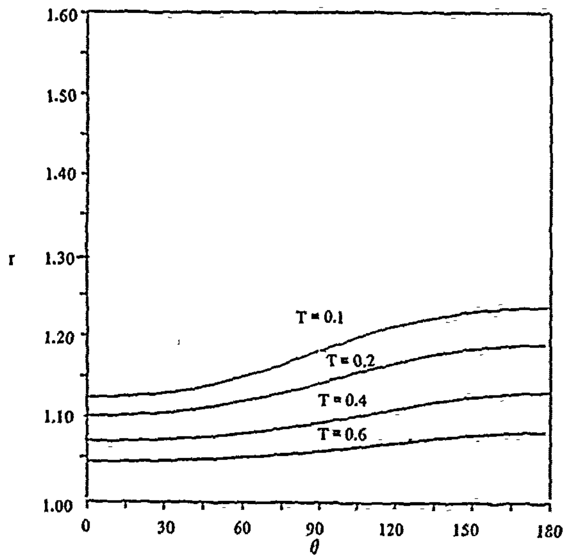


Figure 2. Isothermal for Pr=0.7 at Re=500, Time=0.6.

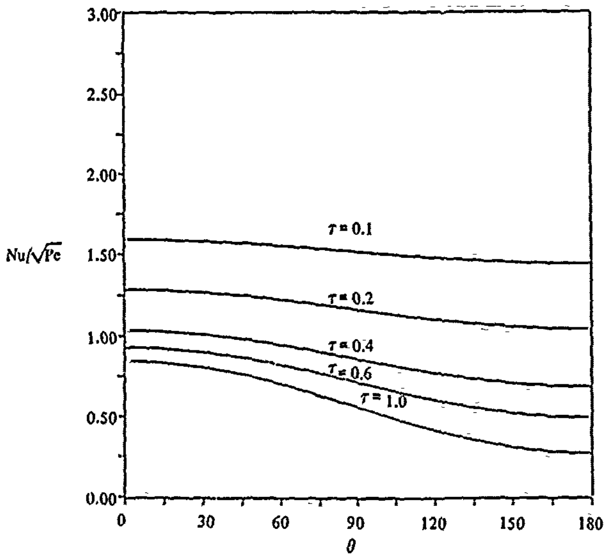


Figure 3(b). Nusselt number distribution on sphere surface for Pr=0.7 at Re=500.

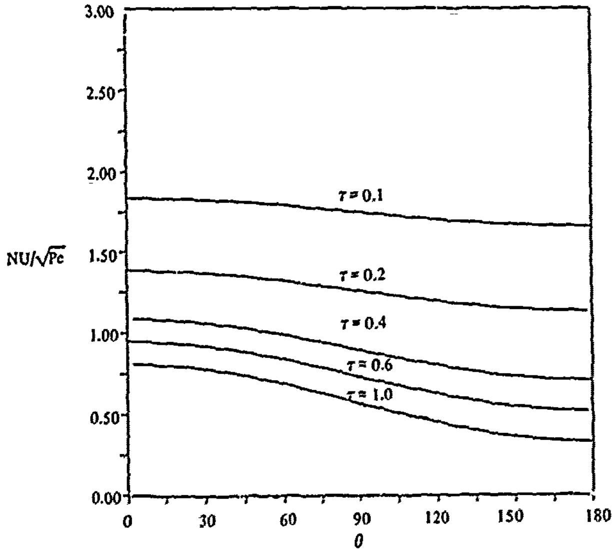


Figure 3(a). Nusselt number distribution on sphere surface for Pr=0.7 at Re=100.

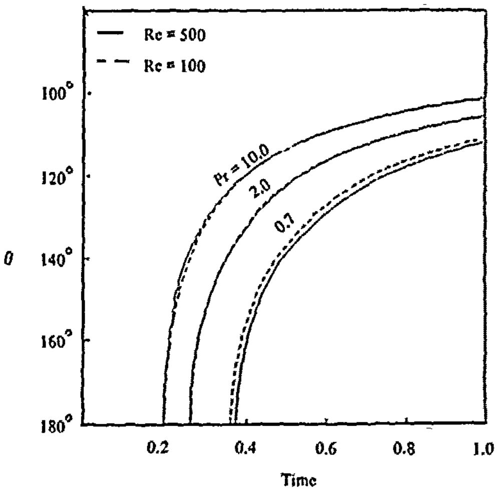


Figure 4. Progression of minimum Nusselt number point with time.

NUMERICAL SIMULATION OF THREE-DIMENSIONAL LAMINAR[1] AND TURBULENT FLOWS OVER BODIES OF ARBITRARY SHAPE[1]

MARKOV A.A.
Institute for Problems in Mechanics
USSR Academy of Sciences
Prospect Vernadskogo 101
117526 Moscow, USSR.

RIZHOV YU.A., SCHEKIN G.A.
Moscow Aviation Institute
Volokolamskoe highway 4
125871 Moscow, USSR.

Abstract- The analysis and generalization of known algebraic models of turbulence have been caried out for three-dimensional boundary layer computations the supersonic flow over finite, twisted wings, taking into account the effects of aerodynamic heating and heat radiation. The results of heat-transfer prediction are in good agreement with experimental data.

Supersonic laminar flows at moderate Reynolds numbers have been simulated on the basis simplified Navier-Stokes equations by space-marching method, combined with global pressure itarations. The generalization of known parabolization techniquc for equations in subsonic regions of the flow and new algorithms of relaxation of the pressure are suggested. The results of three-dimensional shock layer computations over blunt bodies are presented.

I. INTRODUCTION

In recent papers on numerical computation of viscous gas flows it has been shown that the calculation efficiency may be enhanced by using the boundary-layer methods and appropriate scales both in complete Navier-Stokes (N.-S.) equation and simplified N.-S. composite asymptotic equations, namely viscous layer (VL) and viscous layer with azimuth diffusion (VLAD) equations .

The application of boundary layer (BL), VL and VLAD equations and space-marching algorithm along the dominant direction of the flow, combined with the global pressure iteration allows a tremendous reduction in necessary computing time and storage requirements over that required for time - dependent approach for N.-S. equations.

II. LAMINAR FLOWS

We shall write the simplified Navier - Stokes equations in arbitrary curvalinear coordinate system in strictly conservative form.

Let Y_1, Y_2, Y_3 be Cartasian coordinates; X^1, X^2, $X^3 \equiv$ X, Y, Z - curvalinear coordinates of a point M. Let us introduce covariant $\vec{a}$ and contravariant $\vec{a}^i$ basisies:

$$\vec{a}_i = \{ \frac{\partial y_1}{\partial x^i}, \frac{\partial y_2}{\partial x^i}, \frac{\partial y_3}{\partial x^i} \}, \quad \vec{a}^i = \{ \frac{\partial x^i}{\partial y_1}, \frac{\partial x^i}{\partial y_2}, \frac{\partial x^i}{\partial y_3} \}$$

than components g_{ij}, g^{ij} of metric tensor can be expressed as follows:

$$g_{ij} = \vec{a}_i \cdot \vec{a}_j \ ; \quad g^{ij} = \vec{a}^i \cdot \vec{a}^j$$

Let us denote $g = \det\| g_{ij} \|$ and consider vectors $\vec{a}^i$, $\vec{a}_j$ at two points M and M_0, employing index zero for point M_0.

We shall introduce the coefficients G^i_k

$$G^j_k(M,M_0) = \vec{a}^j_0 \cdot \vec{a}_k \ , \text{ than } G^j_k(M_0,M_0) = \delta^j_k$$

We shall use operators L_{VL} and L_{VLAD} for $L_{VL}(f) = 0$, $L_{VLAD}(f) = 0$, where $f = (u^1, u^2, u^3, p, h)^T$, velocity vector $\vec{V} = u^1\vec{a}_1 + u^2\vec{a}_2 + u^3\vec{a}_3 = u^i\vec{a}_i = u_i\vec{a}^i$. The summation on repeated indexes is usualy assumed, if exception is not noted.

$$L(f) = \frac{\partial}{\partial x^j} (A^j(f) - \frac{1}{Re} \frac{\partial}{\partial x^i} (\Phi^{js} \frac{\partial f}{\partial x^s} + \Psi^j))$$

Here j=s=3 for VL and j,s=1,2,3; s=j ≠ 1 for VLAD equations. Components of vectors $A^j = (A^j_1 , \ldots , A^j_5)^T$ and matrices $\|\Phi^{js}\|$ for the point M_0 can be written as follows :

(index zero is omited), $u^1 \equiv u$, $u^2 \equiv v$, $u^3 \equiv w$; $A^1 \equiv A$, $A^2 \equiv B$, $A^3 \equiv C$, $x^1 \equiv x$, $x^2 \equiv y$, $x^3 \equiv z$;

$$A^j_i = G^i_k(pg^{jk} + \rho u^j u^k)\sqrt{g}, \quad i,j,k=1,2,3, \quad p = \frac{(\gamma-1)}{\gamma}\rho h$$

$$A^j_4 = \rho u^j \sqrt{g}, \quad A^j_5 = \rho u^j H\sqrt{g}, \quad H = h+E, \quad E = \frac{1}{2} g_{ij}u^i u^j$$

$$(\Phi^{js})_{im} = \sqrt{g}G^i_k(\bar{\mu}' g^{jk}\delta^s_m + \mu g^{js}G^k_m - \mu g^{js}\delta^k_m + \mu g^{ks}(G^j_m - \delta^j_m))$$

$$(\Phi^{js})_{i4} = (\Phi^{js})_{i5} = (\Phi^{js})_{4N} = 0, \quad i, m=1,2,3, \quad N=1,\ldots,5$$

$$(\Phi^{js})_{5m} = \sqrt{g}(\mu g_{ki}\, g^{js}u^i(G^k_m - \delta^k_m) + \mu u^s(G^j_m - \delta^j_m) + \bar{\mu}' u^j \delta^s_m), \quad (\Phi^{js})_{54} = 0, \quad (\Phi^{js})_{55} = \sqrt{g}\, \frac{\mu}{Pr}\, g^{js};$$

$$\Psi^j_i = \sqrt{g}G^i_k(\bar{\mu}' g^{jk}u^s \frac{\partial \ln\sqrt{g}}{\partial x^s} + \mu g^{js}u^i \frac{\partial G^k_i}{\partial x^s} + \mu g^{ks}u^i \frac{\partial G^j_i}{\partial x^s})$$

$$\Psi^j_4 = 0, \quad \Psi^j_5 = \sqrt{g}(g_{ki}g^{js}u^i u^i \frac{\partial G^k_i}{\partial x^s} + \mu u^s u^i \frac{\partial G^j_i}{\partial x^s}),$$

here $\bar{\mu}'$ is second viscosity coefficient.

We use approximate formulae for components A^1_1, A^1_5 of vector A^1 as follows :

$$A^1_1 = \sqrt{g}G^1_k(\rho u^1 u^k + g^{k1}\chi_1 p + g^{k1}(1-\chi_1)P_g) \qquad (1)$$

$$A^1_5 = \sqrt{g}(\rho u^1(h\chi_2 + E) + \frac{\gamma}{\gamma-1} u^1(1-\chi_2)P_g), \qquad (2)$$

where χ_1, χ_2 are special functions discussed below and P_g is a part of pressure, that is assumed to be known from previous global iteration. If $\chi_1 = \chi_2 = 1$, or $p = P_g$, than formulae become exact. For stability of space-marching algorithm is sufficient, that $u^1 > 0$ and $\chi_1 = \omega_1 F_1(M_1, M_2, \chi_2)$; $\omega_1 < 1$, $\chi_2 > 0$

[1] This work supported by a grant of Soviet-French - Italian joint venture in computer and software buisness 'INTERQUADRO'.

Here $F_1 = 1$, if $M_1 > 1 + \varepsilon$ and

$$F_1 = \frac{\gamma \chi M_1^2 + (\gamma-1) b_1 \ (M_1, M_2)}{\chi_2 + (\gamma-1) \ (M_1{}^2 + b_1)}, \text{ if } M_1 \le 1+\varepsilon, \ 0<\varepsilon \ll 1$$

$$b_j = g_{12} M_1 M_j + g_{j3} M_j M_3 \sqrt{g^{33} g^{jj}}, \ M_j = \sqrt{u^j u^j / (c^2 g^{jj})}$$

$j=1,2,3$ without summation.

The method in question has been tested on computations viscous shock layer over blunt cones and swept infinite wings. The results are in good agreement with known data.

Fig. 1 shows the distributions of shock layer thickness ζ_v (x) and thermal flux coefficient CH(x) along infinite swept wing $R(x) = \sqrt{2x/(1+x)}$ at swept angle 60° for flow $M_\infty = 10$, Re = 50. The results are presented for global iterations of number N = 1, N = 5 (curve 2) and N = 10 (dots). It is sufficient at average N = 10 to obtain the convergence to 10^{-3}.

Figures 2 - 3 refer to the flow $M_\infty = 6$; Re = 3500, H_w=0.35. Fig. 2 shows the distributions of thermal flux CH(x_j, y) and azimuth skin friction coefficient $C_{F2}(x_j, y)$ for sections x_1= 0.83, x_2= 3.43, x_3= 5.19 (lines 1, 2, 3 respectively) along the azimuth coordinate y: $0 \le y \le \pi$. Dash line and dots are presented CH and C_{F2} at x = 0.8 for bielliptic cone $\lambda = 0.1$, where surface of blunt cone has been done by the equation:

$$z = R(x,y); \ R(x,y) = \sqrt{A^2 \sin^2 y + B^2 \cos^2 y}$$

$$A = \cos \alpha_c + (x-x_0) \operatorname{tg} \alpha_c, \ x_0 = 1 - \sin \alpha_c$$

$$B = \cos \alpha_c + (x-x_0) \operatorname{tg} \beta_c l(y),$$

$$l(y) = 1 \text{ for } 0 \le y \le \pi/2, \ l(y) = \lambda \text{ for } \pi/2 \le y \le \pi$$

III. TURBULENT FLOWS

The computations on windward side, slender plane delta wing 65-sweep, sharp leading edges have been carried our for BL equations, including laminar, turbulence transition and turbulent regions of flow. The solution of Euler equations has been obtained by Godunov method.

Effects of turbulence are incorporated by specifying viscosity coefficient according to three turbulence closure models by Spalding, Pletcher and Cebeci with some corrections, that have been found in accordance with experimental data for heat transfer coefficient. The anisotropy of eddy viscosity investigation for Rotta model has been developed, using Klebanoff correction for intermittency at outer vorticity layer and correction for turbulent Prandtl number variation across the boundary layer.

The heat flux distribution over delta wing surface is compared with experimental data[2] Q in Fig. 3. The windward side heat transfer distributions are shown for section $\bar{z}$ =0.25 ($\bar{z} = z/l$; l is semi-span wing size).

The computations were made using turbulence models by Pletcher and Cebeci, including corrections. The results are given at attack angle of 15 for section $\bar{x}$ = 0.33, 0.6, 0.9 (where $\bar{x} = x/c$, c is local wing chord). These sections correspond to region of fully-developed turbulence flow.

The computed q at section $\bar{x}$ = 0.33 is approximately on 5 per cent over and at $\bar{x}$ = 0.9 on 5 per cent lower the corresponding values of Q.

The computed values of q for Cebeci model without correction is on 7 per cent lower, than Q at section $\bar{x}$ = 0.33, but at sections $\bar{x}$ = 0.6, $\bar{x}$ = 0.9 the values q and Q are almost the same. The addition Klebanoff and Rotta corrections to Cebeci model has not changed essentially the values of q. The use Cebeci model with correction of local turbulent Prandtl number gives increasing of q on 2 - 4 per cent. The results of q computation, using Spalding model are approximately on 200 per cent over, than Q.

The analysis of computed results allows us to modify Cebeci model of turbulence. Instead of Clauser formula for outer vorticity layer the following approximation is suggested

$$\mu_t = k_0 \ \rho \ u_{t1} \ \delta^{\bullet}_{3-\Delta}$$

where $\delta^{\bullet}_3$-Δ is boundary layer displacement thickness, k_0 is dimensional constant, that depends on kinematic viscosity coefficient ν_1. The value of ν_1 varries along the wing surface, but this variation is not large so approximation k_0 = const has been used.

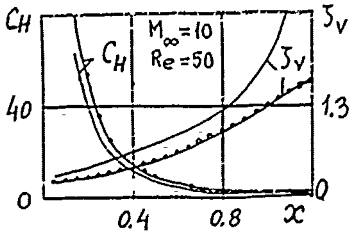


Fig. 1.

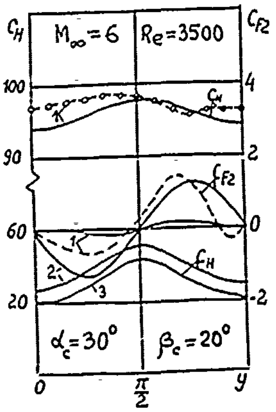


Fig. 2.

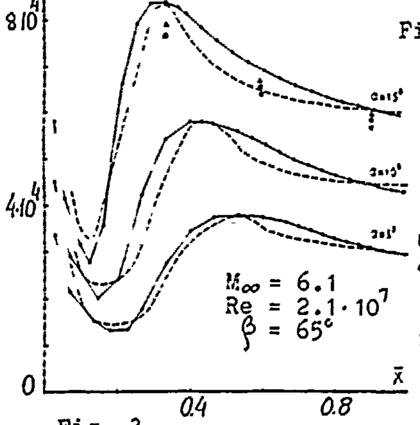


Fig. 3.

[2] Experimental data have been obtained in TzAGI.

THE OPTIMUM SHARE OF HYDROFOIL BENEATH A FREE SURFACE

A.H. Essawy and A.Y.Al-Hawaj

Abstract

The usual assumptions in problems of the obstacle beneath a free surface are taken as a basis: namely, the liquid is non-viscous and moving two-dimensionally, steadily and without voracity, the only force acting on it is gravity. With theses assumptions together with a linearization assumption we determine the forces, due to the hydrofoil to obtain the optimum shape so that the drag is minimum. Analytical solutions by a singular integral equation method, Duhamel's method and some approximate methods are discussed for the linearized theory.

I. Introduction:

There is an extensive literature connected with wave resistance due to a submerged obstacle [see, e.g., Havelock, T.H. (1), (2). Kochin, N.E. (3), Kothcin, N.E. (4), Wehasen, J.V. and LKaition. E.V. (7), Kreisel, G. (5) and Riabouchinsky, D. (6)] but the three papers which are most relevant to present work are those to Kochin , N.E. Kibel, I.A. and Roze, N.V. (3) and Wu, T.Y. and Whitney, A.K. (8) and Essawy, A.H. (9)].

A singular integral equation of the boundary value problem is obtained and can be solved to yield expressions for the lift and drag as functions of the unknown singularity distribution. γ being the vortex strength together with the known shape, z (hydrofoil slope), these expressions are given for a hydrofoil of arbitrary shape.

We use variational calculus technique to evaluate the optimum shape of a two-dimensional hydrofoil of given length and prescribed mean curvature which produces minimum drag.

II. The Hydrofoil Beneath a Free Surface

A hydrofoil of arbitrary shape is in steady, rectilinear motion at a depth h beneath the free surface of a uniform liquid flow with speed U in the x-positive direction.

We assume the liquid is non-viscous and moving two-dimensionally and without vorticity, the only force acting on it is gravity.

The problem will be solved on the basis of linearized theory and for this purpose we introduce the following vortex distribution on the x-axis:

Vortices of strength $\gamma(\xi)$ per unit length in $0<\xi<a$, $y=-h$, $(\gamma>0$, clockwise)

The complex potential due to a single vortex at $(\xi,-h)$ as follows:

$$w(z) = -\frac{\gamma(\xi)}{2\pi i}\{\log(\frac{z-c}{z-\bar{c}}) + 2e^{-i\nu z}\int_{\infty}^{z}\frac{e^{-i\nu t}dt}{t-\bar{c}}\}, \quad [\nu=g/U^2, c=\xi-ih]. \qquad [2.1]$$

The complex potential due to a the complete distribution of vortices as described above will be

$$w(z) = -\int_0^a \frac{\gamma(\xi)}{2\pi i}\{-\log(\frac{z-c}{z-\bar{c}}) - 2\int_0^\infty \frac{e^{-i\nu s}ds}{z+s-\bar{c}}\}d\xi \qquad [2.2]$$

Denoting the potential of this steady motion by

$$W = \Phi + i\Psi \qquad [2.3]$$

Let the x- and y- components if the hydrodynamic forces acting on the hydrofoil be denoted by drag D and lift L, then the complex forces acting on a hydrofoil calculated within the linearized theory are given by

$$D+iL = \int_0^a \{P|_{y=0-} - P|_{y=0+}\}\, idz \qquad [2.4]$$

$$L = -\rho U\int_0^a \gamma(x)dx \ , \ D = \rho U\int_0^a \gamma(x)z(x)\, dx \qquad [2.5]$$

The boundary condition on the hydrofoil it can be approximate to

$$z(x) = \frac{v}{U+u} \simeq \frac{1}{U}v = -\frac{1}{U}\frac{\partial\phi}{\partial y}\Big|_{y=-h} \qquad [2.6]$$

where u,v are the components of liquid velocity as follows:

$$u = -\frac{\partial\Phi}{\partial x} = -\frac{\partial\phi}{\partial x} \ , \ v = -\frac{\partial\Phi}{\partial y} = -\frac{\partial\phi}{\partial y} \qquad [2.7]$$

Using [2.2] we can write [2.6] in the form

$$z(x) = \frac{1}{2\pi U}\int_0^a \gamma(\xi)\{\frac{1}{(x-\xi)} + k(x-\xi)\}d\xi \qquad [2.8]$$

with

$$k(x-\xi) = \frac{(x-\xi)}{(x-\xi)^2+4h^2} - 2\nu\int_0^\infty \frac{[(x-\xi+s)\sin\nu s + 2h\cos\nu s]\,ds}{[(x-\xi+s)^2]} \qquad [2.9]$$

III. The Optimum Shape of Hydrofoil of Minimum Drag

We pose the problem of minimizing the drag coefficient subject to a constraint on curvature K, together with a constraint on the length of the hydrofoil l as following

$$I[\gamma(x),z(x),z'(x),x] = D^* + \lambda_1 l + \lambda_2 K = \int_0^a F[\gamma(x),z(x),z(x),x,\lambda_1,\lambda_2]\,dx \qquad [3.3]$$

with the function $F[\gamma(x),z(x),z'(x),x]$ given by

$$F[\gamma(x),z(x),z'(x),x,\lambda_1,\lambda_2] = \frac{1}{U}z(x)\gamma(x) + \lambda_1\sqrt{1+z^2(x)} + \lambda_2 z'^2(x) \qquad [3.4]$$

where γ, z are related by [2.8] and, λ_1,λ_2 are Lagrange multipliers.

We define an admissible function as any function γ(x) which satisfies the Hölder condition (μ<1) and we define the optimal function as an admissible function which minimize I[γ,z,z',x]

The Necessary Condition Of Optimality. $\delta I[\gamma,\xi]=0$, [3.5]

which yields:

$$z(x) = -\frac{1}{2\pi}\int_0^a [\frac{1}{U}\gamma(s) + \frac{\lambda_1 z(s)}{\sqrt{1+z^2(s)}} - 2\lambda_2 z''(s)][\frac{1}{s-x} + k(s-x)]\,ds = 0 \qquad [3.6]$$

This equation is a necessary condition for th existence of an external I[γ], combines with the integral equation, [2,8], to give a pair of singular integral equations which are to be solved for γ,z subject to appropriate conditions and constraints.

Substituting from [2.8] in [3.6], we obtain

$$\frac{1}{U}\int_0^a \gamma(s)\,[k(s-x)+k(x-s)]\,ds + \int_0^a\left[\frac{\lambda_1 z(s)}{\sqrt{1+z^2(x)}} - 2\lambda_2 z''(s)\right]\left[\frac{1}{s-x} + k(s-x)\right]ds = 0 \qquad [3.7]$$

We consider the solution of [3.7] for the slope z(x) only in the case of small slope, and we approximate to [3.7] as follows:

Now, we use the method of iteration to solve equation [3.8] as follows:

We introduce function sequences of the form

$$z_0, z_1, z_2, \dots\dots, \gamma_0, \gamma_1, \gamma_2, \dots\dots \qquad [3.9]$$

and the stages in the iteration procedure would be as fllows:

(a) First we solve

$$\int_0^a [2\lambda_2 z_0''(s) - \lambda_1 z_0(s)]\frac{ds}{s-x} = 0 \ , \ z_0(x) = \frac{1}{2\pi U}\int_0^a \gamma_0(s)\frac{ds}{s-x} \quad (0<x<a) \qquad [3.10]$$

for γ_0, z_0 which gives

$$z_0(x) = -F_0 m_0 \sin m_0 x + \frac{Co}{2\lambda_2^{(o)} m_0^2}\int_0^x \frac{\sin m_0(x-\tau)}{\sqrt{\tau(a-\tau)}} \qquad [3.11]$$

$$\gamma_0(x) = \frac{2U}{\pi}\sqrt{\frac{a-x}{x}}\int_0^a\sqrt{\frac{s}{a-s}}\frac{z_0(s)ds}{s-x} \qquad [3.12]$$

where F_0, Co is an arbitrary constants and $m_0 = \frac{\lambda_1^{(o)}}{2\lambda_2^{(o)}}$

(b) Secondly, we solve

$$\int_0^a [2\lambda_2 z_1''(s) - \lambda_1 z_1(s)]\frac{ds}{s-x} = \frac{1}{U}\int_0^a \gamma_0(s)\,[k(s-x)+k(x-s)]\,ds - \int_0^a [2\lambda_2 z_0''(s) - \lambda_o z_1(s)]k(s-x)\,dx,$$

$$z_1(x) = -\frac{1}{2\pi U}\int_0^a \frac{\gamma_1(s)ds}{s-x} + \frac{1}{2\pi U}\int_0^a \gamma_0(s)k(x-s)\,ds, \qquad (0<x<a)$$

[3.13]

First we write the inversion of the first equation in [3.13] as follows:

$$2\lambda_2^{(1)}z_1''(x) - \lambda_1^{(1)}z_1(x) = \frac{1}{\sqrt{x(a-x)}}\{c_1 - \frac{1}{\pi^2}\int_0^a \frac{\sqrt{s(a-s)}\phi(s)ds}{s-x} \qquad [3.14]$$

where c_1 is an arbitrary constant and

$$\phi(s) = \frac{2}{\pi}\int_0^a [k(\tau-s) + k(s-\tau)]\sqrt{\frac{a-\tau}{\tau}}d\tau \int_0^a \sqrt{\frac{t}{a-t}}\,[(-F_o m_o \sin m_o t\,)$$

$$+ \frac{E_o}{m_o}\int_0^t \frac{\sin m_o(t-\xi)d\xi}{\sqrt{\xi(a-\xi)}})\,]\frac{dt}{t-\tau} - C_o\int_0^a \frac{k(\tau-s)d\tau}{\sqrt{\tau(a-\tau)}}\ , \quad E_o = \frac{C_o}{2\lambda_2^{(o)}}$$

[3.15]

Equation [3.14] can be written as follows:

$$z_1''(x) + m_1^2 z_1(x) = F(x), \quad [m_1^2 = -\frac{\lambda_1^{(1)}}{2\lambda_2^{(1)}}\ , (0<x<a)], \qquad [3.16]$$

where

$$F(x) = \frac{1}{\sqrt{x(a-x)}}\left\{x\,E_1 - \frac{D_1}{\pi^2}\int_0^a \frac{\sqrt{s(a-s)}\phi(s)ds}{s-x}\right\} [E_1 = \frac{C_1}{2\lambda_2^{(1)}}\ , D_1 = \frac{1}{2\lambda_2^{(1)}}\ , (0<x<a)] \quad [3\ \ .17]$$

it is assumed at this stage $\frac{\lambda_1^{(1)}}{\lambda_2^{(1)}}<0$ and we show later that $\lambda_1^{(1)}<0$, $\lambda_2^{(1)}>0$ are sufficient conditions for a minimization of the drag D. The boundary conditions to be satisfied by $z_1(x)$ are

$$z_1(o) = y_1'(o) = 0, \quad z_1(a) = y_1'(a) = \beta, \quad z_1(0) = 0,$$

$$y_1(0) = 0, \qquad y_1(a) = y_o, \qquad [\beta,\ y_o \text{ prescribed}] \qquad [3.18]$$

The solution of the non-homogeneous differential equation in [3.16] which satisfying the boundary condition (3.18) is

$$z_1(x) = y_1'(x) = -\frac{1}{m_1}\int_0^x F(\xi)\sin m_1(\xi-x)d\xi \ +$$

$$\frac{\sin m_1 x}{m_1 \sin m_1 a}\int_0^a F(\xi)\sin m_1(\xi-a)d\xi + \beta\frac{\sin m_1 x}{\sin m_1 a} \quad (0<x<a) \qquad [3.19]$$

The function $z_1(x)$ in [3.19] should satisy the constraints, and the boundary conditions $z_1'(o)=0$ and $y_1(a)=y_o$; in this way we obtain four unknowns m_1, E_1, D_o, and C_o, which have to b4e evaluated numerically. This problem is resolved numerically in case

$l=4.02$ ft, $a=4$ ft, $k=0.0148$ ft^{-1} $h=16$ ft, $\nu=0.0093$ ft^{-1}

A sufficient condition for the extremum to be a minimum is derived from consideration of the second variation of I is $\delta^2 I>0$ which it can be written in the form

$$\lambda_1 + \frac{2\pi^2}{a^2} + \lambda_2 + \frac{4a}{\pi}\,(\int_0^a \frac{\sin\frac{\pi s}{a}\,ds}{\sqrt{s(a-s)}}\,)^2 > 0.$$

REFERENCES

1. HAVELOCK, T.H. 1932. "The theory of wave resistance". Proc. Roy. Soc.. London, A 138, PP. 339-348.
2. HAVELOCK, T.H. 1926: "The method of images in some problems of surface waves", Proc. Roy. Soc., London. A 115, PP. 268-280.
3. KOCHIN, N.E., KIBEL, I.A. & ROZE, N.V. 1964. "Theoretical hydrodynamics", Interscience publishers.
4. KOTCHIN, N.E. 1951, "On the wave-making resistance and lift of bodies submerged in water". The Society of Naval Architects and Marine Engineers. Trinity Place, New York 6, N.Y.
5. KREISEL, G. 1949. "Surface Waves", Quart. Appl. Math; 7, pp. 21-44.
6. RIABOUCHINSKY, D. 1920. "On steady fluid motions with free surface", Proc. London Math Soc., Vol. 19(2), pp. 206-215.
7. WEHAUSEN, J.V & LAITONE, E.V. 1960. "Surface wave", HandbuchDer-Physick, Vol. Ix, Berlin: Springer-Verlag.
8. Wu, T.Y. & Whitney, A.K. 1973, "Variational Calculus involving singular integral equations", ZAMM, 737-749.
9. ESSWAY, A.H. 1983. "The optimum shape of a noncavitating hydrofoil of maximum lift" Arch. Mech., 35, 2, pp 169-175.
10. Ritger, P.D. & Rose, N.J. 1968. "Differential equation with applications", Mc-Graw-Hill, Inc.

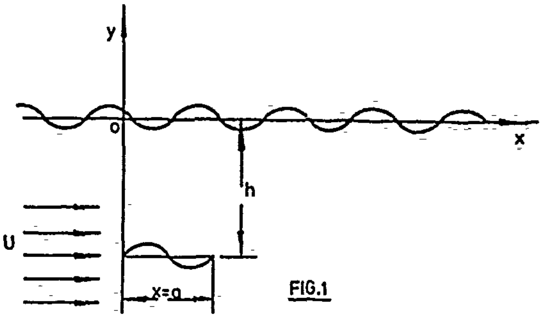


FIG.1

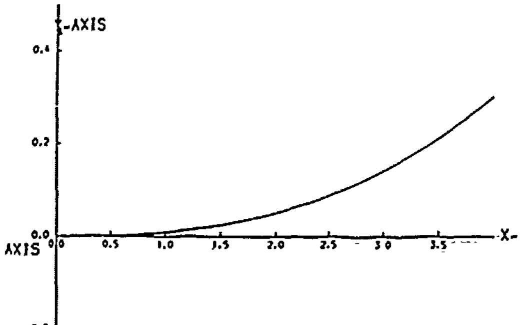


Fig. 2.

A DIRECT SIMULATION OF THE FLOW AROUND A CIRCULAR CYLINDER SINUSOIDALLY OSCILLATING AT LOW KEULEGAN-CARPENTER NUMBERS

Papolu Manikyala Rao, Kunio Kuwahara
The Institute of Space and Astronautical Science,
Sagamihara-Shi,Kanagawa,229,Japan,

and

Kazuhiro Tsuboi
Institute of Computational Fluid Dynamics,1-22-3,
Haramachi,Meguro-Ku,Tokyo,Japan.

ABSTRACT- A finite difference simulation method is presented for the viscous flow field around an arbitrarily moving boundary.Numerical solutions are obtained by directly integrating the incompressible Navier-Stokes equations of finite difference form by adopting a moving grid system,based on a time dependent coordinate transformation.Evolution with time of the flow structures induced by a circular cylinder performing sinusoidal oscillation in a fluid at rest,by means of vortex shedding,is studied at Keulegan-Carpenter number,Kc=9.4.The time dependent drag and lift are also explained.

INTRODUCTION

The study of two dimensional oscillatory flows is of great importance in the design of cylindrical structures such as offshore platforms. Information about the fluctuating forces on an oscillatory cylinder is of special interest to fluid dynamicists and offshore engineers. Several experimental investigations has been carried out on fluid structure interaction problems, a good accounts of which are available in Bearman, et.al(1985), Williamson(1985), Sarpkaya(1986) and Tatsuno and Bearman(1990).

BASIC EQUATIONS AND NUMERICAL METHODS

In the present study,we introduced the following generalized transformations of coordinates,which includes the time variables, in order to deal with moving boundary effectively,

$$\xi^i = \xi^i(x, y, t), i = 1, 2, 3 \qquad (1)$$

where x,y,t are the variables in the physical domain and (ξ^i, $i = 1, 2, 3$) are the variables in the computational domain. The transformation of coordinates and matrix are written as

$$x = x(\xi^1, \xi^2, \xi^3)\ , T_{ij} = \frac{\partial x^i}{\partial \xi^j} J = \det(T^i_j) \qquad (2)$$

where J is the Jacobian.The metric tensor g^{ij} is given by

$$g_{ij} = T^k_i T^l_j \delta_{kl}, g = det(g_{ij}), g^{ij} = (1/2) g^{-1} e^{imn} e^{jpq} g_{mp} g_{nq} \qquad (3)$$

where δ and e denote the Kronecker delta and the Eddington permutation symbol,respectively.

Consider a circular cylinder oscillates in a viscous incompressible fluid,in the direction parallel to x-axis.Instantaneous velocity of oscillation is given by

$$V = V_m \sin(2\pi t/T) \qquad (4)$$

where V_m is amplitude of oscillatory velocity,T period of oscillation and t time.The N-S equations for the sinusoidal flow is written as

$$\frac{1}{Kc}\frac{\partial u^i}{\partial t} + (u^i u^j)_{,j} =$$

$$-g^{ij} P_{,j} + \frac{1}{Re} g^{jk} u^i_{,jk} - \frac{2\pi}{Kc} \cos(2\pi t) g^{ij} T^l_j \qquad (5)$$

$$u^i_{,i} = 0 \qquad (6)$$

The reference scales for non dimensionalizations were $d, V_m, d/V_m, \rho V_m^2$ for the length, velocity, time and pressure, respectively. Dimensionless parameters are Reynolds number $Re = V_m d/\nu$ and $Kc = V_m T/d$. Primitive variables are contravariant components u_i of the flow velocity vector relative to the circular cylinder and pressure P. A subscript with (,) denotes covariant derivative. The last term in the momentum equation(5) represents oscillatory acceleration in the x direction, in which T^i_j is the transformation matrix and g_{ij} is the metric tensor.

The numerical techniques adopted here are based on the well known MAC method,which was orginally developed by Harlow and Welch(1965). The Poisson equation for pressure can be derived on the basis of MAC method.The nonlinear terms are represented by means of a third-order upwind scheme(Kawamura and Kuwahara,1984),e.g.,

$$(U\frac{\partial f}{\partial x})_i = \begin{array}{ll} U_i(f_{i+2} - 2f_{i+1} + 9f_i - 10f_{i-1} + 2f_{i-2})/6h & (U_i > 0 \\ U_i(-2f_{i+2} + 10f_{i+1} - 9f_i + 2f_{i-1} - f_{i-2})/6h & (U_i \leq 0 \end{array} \qquad (7)$$

The poisson equation for the pressure is solved iteratively by employing a modified SOR method.

RESULTS AND DISCUSSION

The computations were performed by a super computer NEC SX-2 (1 3 G flops). The cpu time for a single case ranged from 6 to 10 hs, depending on the value of Re and Kc. The time stepping interval was Δt =0.001. The motions of vortices around a circular cylinder in relative sinusoidal flow are very complicated and the pattern of vortex shedding varies depending on the Kc number and Stokes number,β. Here $\beta = Re/Kc$. Contour maps of vorticity for the case of Re=300,1000;Kc=9.4 are shown in figures(1-2). It may be noted that a pair of small attached vortices are formed behind a cylinder in a starting flow. When the cylinder reverses direction, the attached vortices split up and pair with new vortices in the new half cycle,thus convecting away from the local flow region around the cylinder.This pairing of attached vortices occurs only at the time of flow reversal between small vortices which were still attached to the cylinder just prior to flow reversal. Such a process is shown in figure.1 soon after the begining of cylinder oscillation. The attached eddies are split up as the cylinder moves downwards,and they each form a small vortex pair with a new small vortices. These small vortex pairs formed at flow reversal,where the cylinder reverses again and this eddy pairing repeats itself. The pairing is resonably symmetric initially (Fig.1a) and the attached vortices become unequal in strength and the vortex pairs do not form simultaneously (fig.1b,2) upon flow reversal,giving rise to a lift force of low amplitude fluctuating at the oscillation frequency.This situation is similar to the experimental visulisation of Tatasuno and Berman(1990).The time dependent drag and lift for two different Reynolds numbers are shown in figure.3.

REFEFENCES

Bearman, P.W., Downie, M.J., Graham, J.M.R., and Obasaju, E.D. 1985 Forces on cylinders in viscous oscillatory flow at low Keulegan- Carpenter numbers. J.Fluid Mech. 154, 337-356.

Harlow, F.H. and Welch, J.E.1965 Numerical calculation of Time dependent viscous incompressible flow of fluid with free surface. Phys.Fluids. 8, 2182-2189.

Kawamura, T. and Kuwahara, K.1984 Computations of high Reynolds number flow around a circular cylinder with surface roughness. AIAA Paper 84-0340.

Sarpkaya, T. 1986 Force on circular cylinder in viscous oscillatory flow at low Keulegan-Carpenter numbers. J.Fluid Mech. 165, 61-71.

Tatsuno, M. and Bearman, P.W. 1990 A visual study of the flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers and low Stokes numbers. J.Fluid Mech. 211, 157-182

Williamson, C.H.K. 1985 Sinusoidal flow relative to circular cylinders.J.Fluid Mech. 155, 141-174.

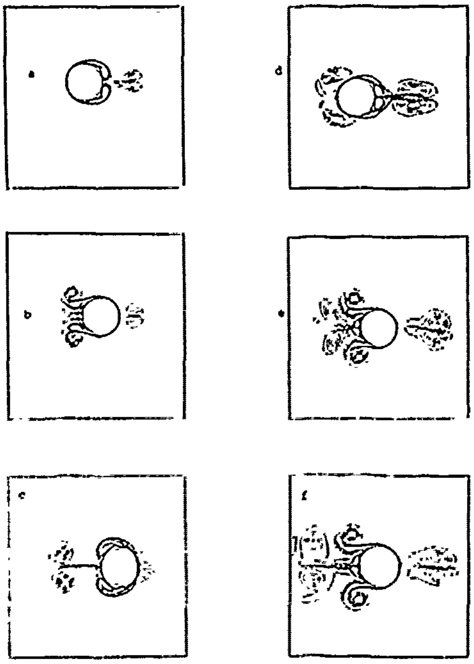


Figure 1a Instantaneous Vorticity contours for the case of Kc=9.4, Re=1000;(a)t=2.0,(b)t=3.0,(c)t=4.0,(d)t=5.0,(e)t=5.0,(f)t=9.

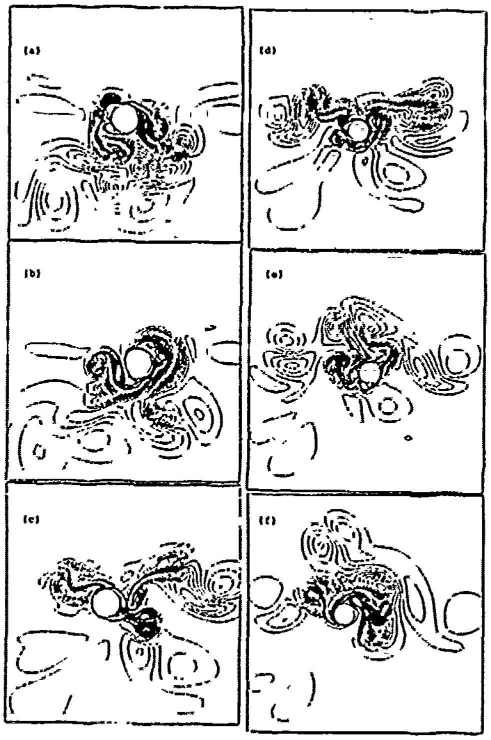


Figure 1(b).As figure 1a),Vorticity contours,(a)T=30.0,(b)t=40.,(c)t=50.0,(d)t=60.0,(e)t=70.0,(f)t=80.0.

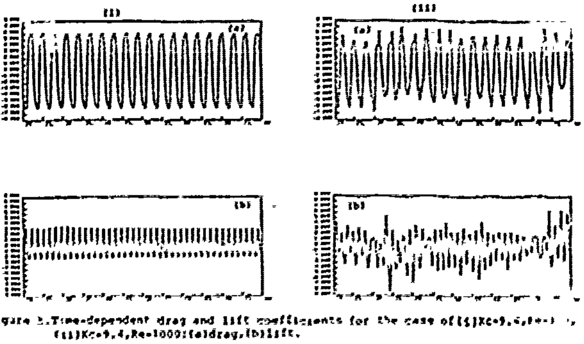


Figure 2.Time-dependent drag and lift coefficients for the case of(i)Kc=9.4,Re=300, (ii)Kc=9.4,Re=1000:(a)drag,(b)lift.

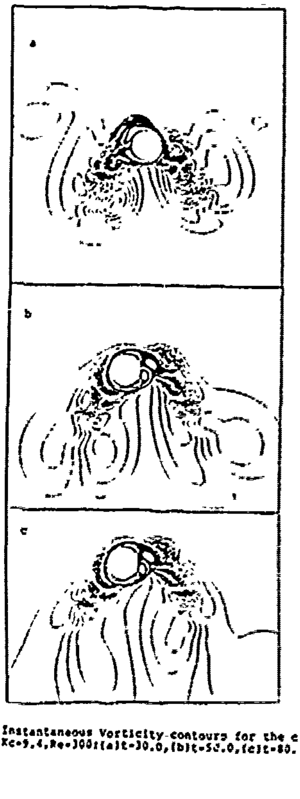


Figure 3.Instantaneous Vorticity-contours for the case of Kc=9.4,Re=300:(a)t=30.0,(b)t=50.0,(c)t=80.0.

EXTENSION OF THE λ-FORMULATION TO IMPERFECT GAS FLOWS

D. LENTINI

Dipartimento di Meccanica e Aeronautica
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18, I-00184 Roma RM, Italy

ABSTRACT

The λ-formulation for compressible flows is extended to gas flows with specific heat varying as a function of the temperature. The proposed formulation is based on a suitably specified function of the temperature, which allows defining the Riemann variables as linear combinations, with constant coefficients, of the dependent quantities. The resulting equations are solved via a fast solution algorithm. A test case is worked out of a nozzle flow, evidentiating the differences with respect to the solution for constant specific heat.

I. INTRODUCTION

The λ-formulation (Moretti 1979, 1987) has proved to be a powerful tool for the numerical solution of compressible flows of perfect gas. For 'perfect' we mean that the gas is assumed to be both thermically and calorically perfect. The first attribute refers to its obeying the perfect gas law, while the latter denotes that its specific heat c_p is taken as a constant.

This formulation has been successfully extended to finite-rate chemically reacting flows (Lentini and Onofri 1986, 1987 A and B), but still with the limitation that the component gases of the reacting mixture are perfect.

While the assumption of thermally perfect gas is closely approximated in virtually all cases of practical interest, the requirement that the gas (or the component gases) are calorically perfect may be unsatisfactory to describe flows subjected to large temperature excursions. Indeed, c_p exhibits a fairly large variation with temperature. For example, the specific heat at constant pressure of air varies by about 25% in the temperature range $300 - 2000°K$.

In this paper we extend the λ-formulation to (inert) flows of thermally perfect gas with c_p varying as a function of the temperature T. Such gases are sometimes referred to as imperfect gases. The resulting formulation is extremely simple and involves a limited computational overhead over the perfect gas case.

It is applied here to the computation of the flow in a quasi one-dimensional nozzle in order to prove the workability of this approach and to evidentiate the differences with respect to a perfect gas computation. However, it will be apparent that the range of application of the present formulation is completely general.

II. FORMULATION

The formulation is presented here for simplicity for a quasi one-dimensional flow.

We assume as state variables the speed of sound a and the entropy s, and the velocity u as the motion variable. Consequently, the continuity and momentum equations, which we write for convenience in the form

$$a\,\frac{\rho_t + u\,\rho_x}{\rho} + a\,u_x = -a\,u\,\frac{A_x}{A} \tag{1}$$

$$u_t + u\,u_x + \frac{1}{\rho}\,p_x = 0 \tag{2}$$

will be recast in terms of the variables u, a, s. To this end, we observe that the speed of sound is related to the temperature T via the relationship

$$a^2 = \gamma\,R\,T \tag{3}$$

γ being the gas specific heats ratio and R its constant Upon logarithmic differentiation we obtain

$$2\,\frac{a'}{a} = \left(1 + \frac{\gamma_T\,T}{\gamma}\right)\frac{T'}{T} \tag{4}$$

Here the prime denotes differentiation with respect to either t or x, γ_T is the derivative of γ with respect to T. From the first principle of thermodynamics, written for adiabatic flows with pressure work only

$$c_v\,T' = R\,T\,\frac{\rho'}{\rho} \tag{5}$$

one gets, in view of (4)

$$\frac{\rho'}{\rho} = \frac{1}{\delta\left(1 + \frac{\gamma_T T}{\gamma}\right)}\,\frac{a'}{a}$$

with the position $\delta = (\gamma - 1)/2$. This expression will be substituted in the continuity equation.

The pressure gradient term in the momentum equation can be expressed by means of the thermodinamic relationship

$$\frac{1}{\rho}\,p' = h' - T s'$$

where h is the enthalpy; then, after eq. (4), with $h' = c_p\,T'$

$$\frac{1}{\rho}\,p_x = \frac{1}{\delta\left(1 + \frac{\gamma_T T}{\gamma}\right)}\,a\,a_x - T s_x$$

The set of eqs. (1, 2) can then be recast as

$$\frac{1}{\delta\left(1 + \frac{\gamma_T T}{\gamma}\right)}\,(a_t + u\,a_x) + a\,u_x = -a\,u\,\frac{A_x}{A} \tag{6}$$

$$u_t + u\,u_x + \frac{1}{\delta\left(1 + \frac{\gamma_T T}{\gamma}\right)}\,a\,a_x - T\,s_x = 0 \tag{7}$$

At this juncture, we make the crucial remarks that the terms a' can be expressed as $a_T\,T'$, being a a function of the temperature T, and that the term

$$\frac{1}{\delta\left(1 + \frac{\gamma_T T}{\gamma}\right)}\,a_T = \frac{1}{\gamma - 1}\sqrt{\frac{\gamma R}{T}} = \frac{c_p}{a}$$

is solely a function of the temperature as well. We can thus define the function

$$F(T) = \int_{T^0}^{T} \frac{c_p}{a}\,d\theta \tag{8}$$

where θ is the temperature as a running integration variable, and T^0 is an arbitrary reference temperature. Notice that the integrand is always positive so that the function $F(T)$ is monotonic and can easily be inverted. This definition allows expressing the terms in the derivative of a in eqs. (6,7) as

$$\frac{1}{\delta\left(1 + \frac{\gamma_T T}{\gamma}\right)}\,a_t = F_t$$

and similarly for the derivative with respect to x. We can then recast eqs. (6,7) in terms of the new variable

$$F_t + u F_x + a u_x = \beta \qquad (9)$$

$$u_t + u u_x + a F_x - T s_x = 0 \qquad (10)$$

with $\beta = -a u A_x / A$. By summing and subtracting eqs. (9,10), and with

$$\lambda_1 = u + a \qquad \lambda_2 = u - a$$

we get the final form

$$R_{1t} + \lambda_1 R_{1x} - T s_x = \beta \qquad (11)$$

$$R_{2t} + \lambda_2 R_{2x} + T s_x = \beta \qquad (12)$$

having defined the Riemann variables as

$$R_1 = F + u \qquad R_2 = F - u$$

Thus, the new variable F allows the definition of Riemann variables for imperfect gas flows as linear combinations, with *constant* coefficients, of the dependent quantities, in analogy with the formulation for perfect gas. In the case of isentropic, strictly one-dimensional flow, such variables represent true Riemann invariants.

Further, the imperfect gas formulation becomes formally identical to the perfect gas one.

It is apparent that the present formulation can be extended without any difficulty to multidimensional flows, by redefining the Riemann variables in a similar fashion.

III. SOLUTION ALGORITHM

As an example of the present formulation, here we apply it to the flow in a converging-diverging nozzle. We limit our analysis to the isentropic case for the sake of simplicity, and further consider the steady state solution only.

A semi-implicit algorithm, developed along the guidelines of Moretti's (1983) fast solver, is used. In this iterative technique eqs. (11,12) are integrated separately, at each step, by successive sweeps all over the computational domain.

The discretized form of eq. (11) is, with second-order upwind differencing:

$$\frac{R_{1,n} - \hat{R}_{1,n}}{\Delta t} + \overline{\lambda}_1 \frac{R_{1,n} - R_{1,n-1}}{\Delta x} = \overline{\beta}$$

being λ_1 always positive in the flow under consideration; the averages are defined as

$$\overline{\lambda}_1 = \frac{\lambda_{1,n} + \lambda_{1,n-1}}{2} \qquad \overline{\beta} = \frac{\beta_n + \beta_{n-1}}{2}$$

The caret denotes the previous iteration level.

As far as eq. (12) is concerned, we have to make a distinction between the cases $\lambda_2 > 0$ and $\lambda_2 < 0$. In the former case (supersonic flow) the discretization is analogous to that for R_1:

$$\frac{R_{2,n} - \hat{R}_{2,n}}{\Delta t} + \overline{\lambda}_2 \frac{R_{2,n} - R_{2,n-1}}{\Delta x} = \overline{\beta}$$

with

$$\overline{\lambda}_2 = \frac{\lambda_{2,n} + \lambda_{2,n-1}}{2} \qquad \overline{\beta} = \frac{\beta_n + \beta_{n-1}}{2}$$

whereas for $\lambda_2 < 0$ (subsonic flow) we get

$$\frac{R_{2,n} - \hat{R}_{2,n}}{\Delta t} + \overline{\lambda}_2 \frac{R_{2,n+1} - R_{2,n}}{\Delta x} = \overline{\beta}$$

with

$$\overline{\lambda}_2 = \frac{\lambda_{2,n} + \lambda_{2,n+1}}{2} \qquad \overline{\beta} = \frac{\beta_n + \beta_{n+1}}{2}$$

Astride the sonic line the accuracy of the algorithm is reduced to first-order, by setting

$$\overline{\lambda}_1 = \lambda_{1,n} \qquad \overline{\lambda}_2 = \lambda_{2,n} \qquad \overline{\beta} = \beta_n$$

in order not to violate the domains of dependance of the variables. The eq. in R_1 is integrated by sweeping in the positive x-direction, whereas the one in R_2 is integrated by sweeping from the sonic line to the exit (supersonic region) and from the sonic line to the inlet (subsonic region). This algorithm gives a very fast convergence to the steady solution.

For the case of imperfect gas, the procedure is initialized by first guessing the temperature field, and accordingly estimating F via the relationship

$$F = F(T) \qquad (13)$$

as in (8). The a field is analogously initialized by means of the relationship

$$a = a(T) \qquad (14)$$

as in (3). This allows computing λ_1 and λ_2.

Then, at each iteration step the new values of R_1 and R_2 are computed, and u and F are updated as

$$u = \frac{R_1 - R_2}{2} \qquad F = \frac{R_1 + R_2}{2}$$

The speed of sound is then recomputed as

$$a = a(F) \qquad (15)$$

where the argument of (14) has been transformed into F by inverting eq. (13). λ_1 and λ_2 can then be recomputed and a new iteration cycle started, until convergence is attained.

The boundary condition at the inlet involves matching the total enthalpy

$$h_0 = h + \frac{u^2}{2}$$

where h is recovered as

$$h = h(F) \qquad (16)$$

with the stagnation enthalpy h_c, i.e.:

$$h_c = h(R_2 + u) + \frac{u^2}{2}$$

where F is expressed via the value of R_2, computed from downstream. The above nonlinear relationship is iterated to get the value of u at the inlet.

Once the converged solution is obtained, the density can be recovered as a function of F, via eq. (5):

$$log \frac{\rho}{\rho_0} = \frac{1}{R} \int_{T_0}^{T} \frac{c_v}{\theta} d\theta \qquad (17)$$

the pressure is then computed by means of the (thermal) equation of state.

In the present implementation the functions defined by eqs. (13-17) are computed off-line and approximated by 4th-order fits obtained by means of orthogonal polynomials, in order to make the computational procedure as straightforward as possible.

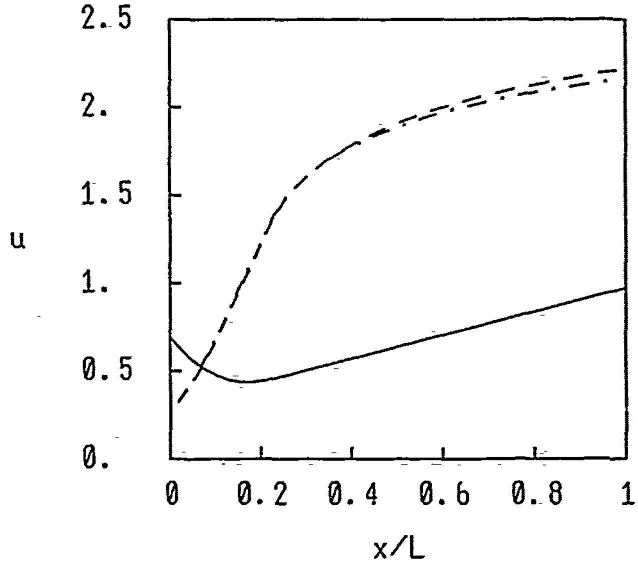


Fig. 1. Axial profiles of velocity u (made dimensionless with the reference value $\sqrt{RT_c}$). Dash-dotted line, perfect gas, dashed line, imperfect gas; solid line, nozzle contour.

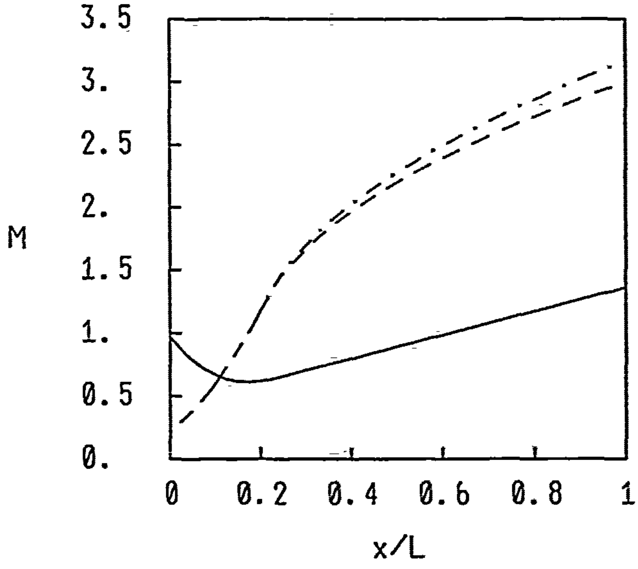


Fig. 2. Axial profiles of Mach number M. Dash-dotted line, perfect gas; dashed line, imperfect gas; solid line, nozzle contour.

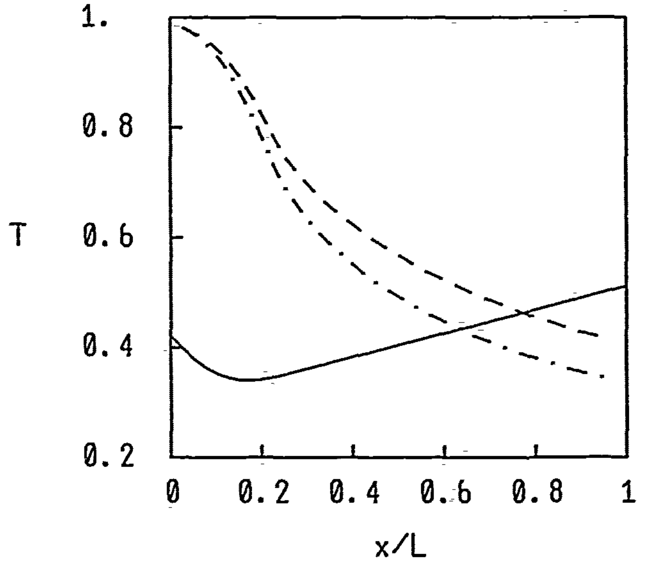


Fig. 3. Axial profiles of temperature T (made dimensionless with the reference value T_c). Dash-dotted line, perfect gas, dashed line, imperfect gas; solid line, nozzle contour.

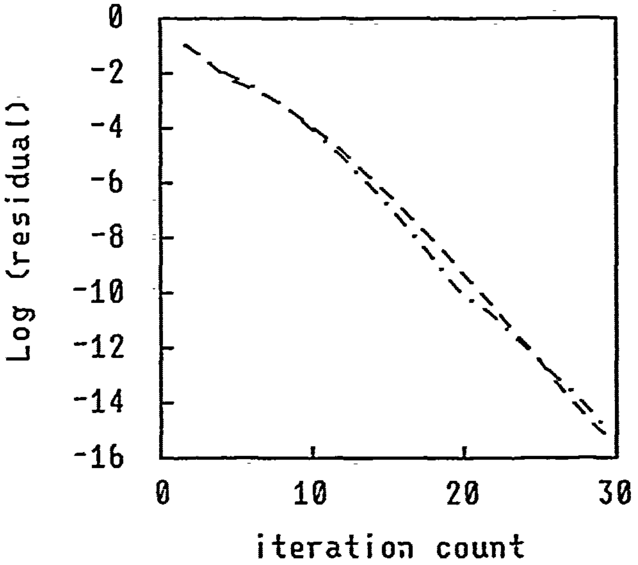


Fig. 4. Decay of the residual of the computation. Dash-dotted line, perfect gas; dashed line, imperfect gas.

IV. RESULTS

We show here a comparison between a calculation performed for perfect gas (with $\gamma = 1.4$) and the present approach for imperfect gas, for a critical flow in a converging-diverging nozzle. The working fluid is assumed to be air, with caloric properties as a function of the temperature accounted for by means of the so-called NASA polynomials (e.g. see Gardiner 1984).

A nozzle with conical converging (semi-angle 45°) and diverging (semi-angle 15°) sections, matched by a throat section with a circular profile, is considered. The inlet and throat radii are 0.19 and 0.12, respectively, the radius of curvature of the throat is equal to the trhoat radius, and the geometric expansion ratio (exit to throat area) is 5. The stagnation temperature T_c is assumed to be 2000°K (note that for imperfect gases results do not scale with temperature). Although at this high temperature some molecular dissociation and formation of nitric oxides do occur, a chemical equilibrium computation (performed with the code by Reynolds 1981) shows that the cumulative mass fraction of the ensuing products is less than 1%, and accordingly in this study we neglect effects related to the varying composition. In particular, the variation in the average molecular weight turns out to be absolutely negligible.

A computational grid with 40 nodes is chosen as, for perfect gas, it gives a relative discrepancy between the Mach number computed analytically and by the numerical solution at convergence limited to a maximum (all over the computational domain) of 0.7%.

Figure 1 compares axial profiles of velocity u for the two cases of perfect and imperfect gas. The nozzle profile is superimposed on the figure.

In the same fashion Fig. 2 compares axial profiles of Mach number, and Fig. 3 profiles of temperature.
It can be observed that while [illegible] and $\dot{M}$ are affected to a limited extent by the varying c_p, the temperature computed with the imperfect gas model exhibits a much slower decay than the perfect gas solution, owing to the higher specific heat. In particular, the gap between the two solutions at the outlet is about $200°K$, thus underlining the need to account correctly for the caloric properties of the gas.
Further, neglecting the effects of temperature on c_p leads to overstimating the mass flow rate by 4.1%, and the velocity thrust by 1.5%; accordingly, the specific impulse is understimated by 2.5%.
Fig. 4 compares the convergence history of the computations for perfect and imperfect gas. The solution is initialized by prescribing the same (arbitrary) Mach number distribution in both cases. It is apparent that the solution algorithm described in Sect. III gives an extremely fast convergence rate, with the steady state reached in as few as 30 iterations.

V. CONCLUSIONS

A simple formulation to effectively extend the λ-formulation to imperfect gas flows is presented and tested.

REFERENCES

Gardiner W.C. (ed.) (1984), *Combustion chemistry*, Appendix C, Springer-Verlag, New York.

Lentini D., Onofri M. (1986), "Solutore numerico veloce per flussi chimicamente reagenti in nonequilibrio", L'Aerotecnica Missili e Spazio, Vol. 65, no. 3.

Lentini D., Onofri M. (1987 A), "Nonequilibrium chemically reacting flows in nozzles", in *Analysis and Design of Advanced Energy Systems: Fundamentals*, M.J. Moran and R.A. Gaggioli eds., ASME AES Vol. 3-1, New York.

Lentini D., Onofri M. (1987 B), "Fast numerical technique for nozzle flows with finite-rate chemical kinetics", in *Computational Fluid Dynamics*, Proceedings II International Symposium on Computational Fluid Dynamics, Sydney.

Moretti G. (1979), "The λ-scheme", Computers & Fluids, Vol. 7.

Moretti G. (1983), "A fast Euler solver for one-dimensional flows", NASA CR 3689.

Moretti G. (1987), "An efficient Euler solver, with many applications", AIAA-87-0352.

Reynolds W.C. (1981), STANJAN, Stanford University.

The perfect-absorbing boundary conditions for the approximate hydrodynamics models.

Novikov V.A., Fedotova Z.I.

Comp.Center,USSR Academy of Scinces
Siberian Division,Krasnoyarsk

Comp.Center,USSR Academy of Scinces
Siberian Division,Krasnoyarsk

The construction method of the perfect-absorbing boundary conditions on two domains boundary, on one of which the fluid motion is described by linear, and in another one-by nonlinear shallow water equations, have been described. The approach proposed also permitts to solve the problem on a free wave pass through the outer boundary out of the domain and the problem on the reflected waves outgoing from the domain through the boundary, on which the disturbances are preset as boundary conditions.

1. With numerical simulating of long waves propagation in water areas one often deals with the following, at the first sight "different" problems:

Problem 1. Let's describe the long waves propagation from the source to the shore. It's known, that non-collapsing wave motion along the deep ocean is rather well described by the linear shallow water equations, whereas at the wave outgoing to the shallows, where non-linearity effects become significant, the non-linear equations should be applied. While solving this problem it's expedient to divide all the flow region into two subregions and on the line, dividing them, to preset the boundary conditions, passing the wave from the "deepwater" region to the "coastal" one without reflection. It's clear, that the question is only about non-physical reflections, caused by different models "joint".

Problem 2. Let's describe the wave motion in the bay, caused by the fact, that on the boundary, faced to the sea, the wave train is preset as the boundary condition. Suppose, that the bay bottom is such, that the reflected wave, going to meet the bay entering one, begins to form rather early. Thus, here the reflected wave out of the region through the boundary, on which the disturbance is assigned.

Problem 3. With the application of finite-difference methods for the wave processes simulation in the open water reservoirs the calculation is performed in the finite region. Thus, here occurs the problem on the free pass of the waves, coming to the calculated region boundary. The boundaries, possessing the property to pass any disturbances wuthout reflection, we should call "transparent" or "absorbing". These boundaries have no physical essence and their consideration is connected with the method of the numerical calculations performance.

In spite of the fact, that any computer algorithm for hydrodynamics problems solution includes the conditions on one or another "transparent" boundaries, these questions are poorly treated in literature.

The most classical is the problem 3. The approaches to its solution are cosidered in the works [1], [2], [3] and in the number of others. The Sommerfeld condition is generally applied [1]. The most close to ours is the approach, represented in [2], [3], devoted to the conditions on the outer boundaries in gas-dynamics problems.

In the present paper the approach, equally applicable for the problems 1-3 solution, has been proposed. This approach works in the cases, when one successes in paravariables introduction, which,firstly,are perfectly connected with the medium physical meters, and, secondly, describe the disturbances, moving in the definite direction. In the case of one-dimensional hyperbolic equations set Riman invariants are the best for such an aim. The arises the problem on the complete outgoing of one-dimensional and two-dimensional problems.

2. We describe the obtained boundary conditions. illustrating, for convenience, the problem geometry and shallow water equations by the fig. 1.

Riman invariants continuity requirement at the point $x=x_*$ can be written as the following conditions on the boundary $x=x$:

$$u_1=u_2,$$

$$\zeta_1=2[d_1(d_2+\zeta_2)]^{1/2}-2(d_2d_1)^{1/2},$$

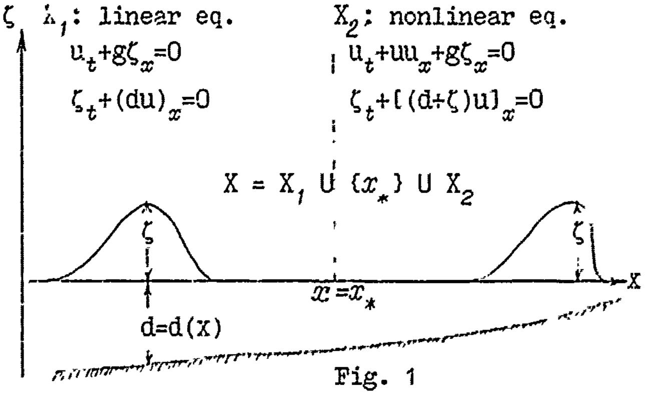


Fig. 1

These formulas are considered to be the basis for finite-difference algorithms construction while solving the problem about the solution joint on two mediums boundary. To construct the boundary conditions for the problems of the complete wave outgoing through the outer boundary the auxiliary technique has been applied. Its main point is the extension of the water filled domain by the channel with constant depth bottom, where the flow is described by linear shallow water equations. This procedure results in the following boundary conditions at the point $x=x_*$, where $x=x_*$ is the left domain boundary:

$$u|_{x=x_*} = 1/2(r+s) \;,$$
$$\zeta|_{x=x_*} = [r-s+4(gd_*)^{1/2}]^2/(16g) - d_* \;,$$
$$r=2(g/d_*)^{1/2}\tilde{\zeta} \;, \qquad d_*=d|_{x=x^*},$$

$\tilde{\zeta}=\zeta(t)$ -is the height of the wave, given on the input, $s=s(x_*,t)$-is the value of the s-invariant , describing the reflected wave, at worked out approach allows to solve both the point $x=x_*$. For the case of the free pass problem the boundary conditions on the boundary $x=x_*$ situated on the right of domain,assume the form:

$$u|_{x=x_*} = r/2,$$
$$\zeta|_{x=x_*} = [r+4(gd_*)^{1/2}]^2/(16g)-d_*, d_*=d|_{x=x_*},$$

where $r=r(x_*,t)$ is r-invariant,describing the wave, which will outgo out of domain bounds.

The main point in the corresponding numerical algorithms is r- and s-invariants approximate construction in boundaries neighborhood.

During the construction of analogous boundary conditions for the two-dimensional problem the assumption of the possibility of such approximate flow description,that in the limits of the small discrete time intervals the wave "front" should be straight and keep its motion direction, was made. For this case the application of one-dimensional algorithm. worked out without difficulties, for the two-dimensional problems was a success.

The series test calculations, demonstrating the workability of the method proposed and a good accuracy of the numerical algorithms, has been performed. Fig.2,3 illustrate the solutions of the problem on the wave outgoing out of the domain. The waves surfaces for the sequential time moments have been shown.

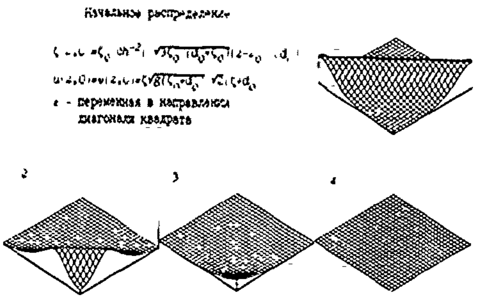


Fig. 2

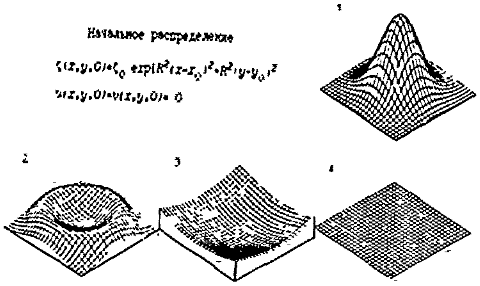


Fig. 3

1.Engquist B.,Majda A. Absorbing Boundary Condition for the Numerical Simulation of Waves//J. Math. Comp.-I977.-V. 3I.-629-65I.
2.Hedstrom G.W. Nonreflecting Boundary Condition for Nonlinear Hyperbolic Systems//J. Comput. Phys.-I979.-V. 30.-222-237.
3.Thompson K.W. Time Dependent Boundary Conditions for Hyperbolic Systems//J. Comput. Phys.-I987.-V. 68.-I-24.

A VORTEX METHOD FOR BLUFF BODY FLOWS AT LOW REYNOLDS NUMBER

A.P.BURROWS AND P.G.BELLAMY-KNIGHTS
Department of Engineering
University of Manchester
Manchester, M13 9PL, U.K.

Abstract-The impulsively started flow past a circular cylinder in a uniform freestream for Reynolds numbers. Re from 200 to 500 is computed by a time splitting method to get the lift. drag. and Strouhal number.

1. INTRODUCTION

The discrete vortex method is a long established approach for modelling the convection of vorticity in two-dimensional problems of separated flow past bluff bodies. See Gerrard[1] and Sarpkaya[2]. This method. which can now also take diffusion into account. has recently been extensively reviewed by Sarpkaya[3]. In the time-splitting method. the processes of convection and diffusion are treated separately. Whereas Chorin[4]. and Smith and Stansby[5] use random walk methods to model the diffusion, Benson et al.[6] introduced and Burrows[7] developed a vorticity re-distribution method in an attempt to reduce the number of discrete vortices required by the model and hence reduce the magnitude of the computational task. In [6]. after extensively testing the parameters of the model at Re=40 (based on diameter). the method was applied to flow past circular cylinders for Re=20. 40. 100. and 200. The present work describes further development and validation of this approach for Re up to 500.

2.MATHEMATICAL AND COMPUTATIONAL FORMULATION.

The Navier-Stokes equations for an incompressible fluid are expressed in non-dimensional form in terms of the vorticity. ζ. and the stream function. ψ .as follows

$$\frac{\partial\zeta}{\partial t} + (\underset{\sim}{u}\cdot\nabla)\zeta = \frac{2}{Re}\nabla^2\zeta \qquad (1)$$

$$\nabla^2\psi = -\zeta \qquad (2)$$

where t is the time and $\underline{u}$=(u.v) are the velocity components in (x.y) Cartesian coordinates with origin at the centre of the cylinder and the x-axis in the freestream direction. Assuming ζ is known at a typical instant. equation (2) is solved for ψ as follows. A radially expanding cylindrical polar mesh (R(j).TH(i)) about the cylinder is transformed into a uniform rectangular mesh (TH(i).RD(j)) by the equation

$$R(j)=\exp(C1.RD(j)/DRD) \qquad (3)$$

where RD(1)=0.

$$RD(j)=RD(j-1)+DRD \quad (j=2.3.....NR) \qquad (4)$$

There are also NR mesh points in the i-direction of length DTH. where

$$DTH=2\pi/(NR-1). \qquad (5)$$

Then the transformed equation (2) is solved for ψ on the rectangular mesh using the Poisson solver of Le Bail[8]. which requires that NR is of the form 2^n+1. NR=129 is found to be a good compromise between resolution and economy. After choosing the radial extent of the flowfield. R(NR)=30. say. (3) gives C1=0.03423. R(NR) must be sufficiently large to ensure that the grid always contains all the vorticity for the duration of the computation. DRD is a constant chosen to be the same order as DTH. Next. the velocity field is computed. Then the known vorticity field is discretised onto point vortices which are convected by Eulerian integration for a small convection timestep. DTT. This procedure is Reynolds number independent and so the mesh and other computational parameters should be invariant as Re changes.

While convecting. it is assumed that these point vortices are diffusing as Oseen vortices. After a time t. a zero age Oseen vortex will diffuse a radial distance $4.4836\sqrt{2t/Re}$. (see Slaouti[9]) and so as the vortex spreads. each part of it will move with an increasingly different velocity and so become distorted. To overcome this problem the diffused vorticity of each vortex is re-discretised onto new zero-age point vortices every timestep. DTR. The validity of this approach was appraised in test cases described in [6] and is satisfactory if DTR is large enough to allow vorticity to diffuse at least two mesh lengths. For a square mesh of length GS. this gives

$$DTR > 0.0995.Re.GS^2 \qquad (6)$$

where Slaouti's result above has been used.

The vorticity in the flowfield is located in two main areas. Firstly. a wake region extends downstream of the cylinder. This is conveniently embraced within a grid of width -5<y<5. length -3<x<R(NR) with square mesh of length GS. Here. GS=0.2. If. for example. Re=200. then (6) implies a minimum diffusion timestep of 0.8 in this mesh.

Secondly. there is a boundary layer region of high vorticity around the cylinder of approximate thickness BL. say at the shoulders of the cylinder where

$$BL = 3.82/\sqrt{Re} \qquad (7)$$

(See [7]). When. for example. Re=200. BL=0.27 and when Re=500. BL decreases to 0.17. Although this region is contained within the rectangular mesh. a finer body fitted mesh allowing more resolution of the diffusion near the surface and a diffusion timestep. DTI say. lower than for the rectangular mesh is required in order to calculate the aerodynamic forces on the cylinder. In [6]. the polar mesh defined for the convection is also used to apply the redistribution algorithm near the body. This grid has mesh size 0.034 X 0.049 on the body expanding to 0.042 X 0.062 at radius 1.27. Now in timestep DTR. vortices in the courser rectangular mesh may diffuse through the finer polar mesh to the surface of the body. To prevent this. all vortices within radius 1.4 are diffused using the fine polar mesh. Since for Re<200 there are at least 8 mesh points within BL. the convection mesh satisfactorily duplicates to treat the diffusion. For Re=500. however. there are only 5 mesh points within BL and so a new and separate polar mesh for diffusion near the body is now introduced. It has the same form as equations (3.4.5) but with DTH halved. For equal mesh lengths in the radial and circumferential directions, C1 is reduced to C1=0.024034. This gives more mesh points within BL. Then for Re≤500 there are at least 8 mesh points within BL. Then DTI has a minimum value of Re/10^4. This is calculated from (6) based on the maximum mesh length at radius 1.4 in the polar grid.

Now the cost of the diffusion algorithm increases as the square of the number of mesh points a vortex diffuses in a timestep. For a fixed mesh, this limits the magnitude of the timestep. For the rectangular mesh, DTR is chosen so that a vortex diffuses through no more than 3 mesh lengths. Then

$$DTR < 0.2239.Re.GS^2 \qquad (8)$$

Similarly, for the polar diffusion mesh, the timestep is limited so that vortices diffuse through no more than 3 mesh lengths at radius 1.4. Now the mesh length is smaller on the cylinder surface and so for the above restricted timestep, the algorithm must actually allow diffusion over 5 mesh lengths near the surface.

Methods of satisfying the no-slip condition on the boundary, relecting vorticity from the surface of the cylinder, calculating the lift and drag coefficients and other details are described in [6] and [7].

3.VALIDATION AND DISCUSSION OF THE RESULTS.

Table 1 shows 6 different sets of parameters for which the computer code was run for Re=200. Their choice was guided by the above considerations. NTH2 is the number of mesh points in the circumferential direction for the inner polar diffusion mesh.

TABLE 1. PARAMETERS OF THE MODEL

Version	DTT	DTI	DTR	NTH2
R200V1	0.2	0.2	1.0	257
R200V2	0.075	0.075	0.975	257
R200V3	0.1	0.2	1.0	129
R200V4	0.075	0.075	1.95	257
R200V5	0.05	0.1	1.0	129
R200V6	0.05	0.1	2.0	129
R250	0.05	0.15	1.95	129
R400	0.075	0.075	1.95	257
R500	0.075	0.075	2.475	257

The diffusion in the inner polar mesh takes place every one or two convection timesteps whereas the diffusion in the outer rectangular mesh is much less frequent due to the larger mesh length. For the range of Reynolds number investigated, periodic oscillating wakes are obtained. The Strouhal number, St, average drag coefficient, Cd(av) and range (twice the amplitude) of the drag, Cd(ra) and lift Cl(ra) coefficients are computed. These are shown in Table 2, with results of Brazer et al.[10] for Re=200.

TABLE 2. COMPUTED RESULTS

Version	Cd(av)	Cd(ra)	Cl(ra)	St
R200V1	1.45	0.10	1.46	0.194
R200V2	1.37	0.13	1.58	0.190
R200V3	1.38	0.11	1.55	0.190
R200V4	1.37	0.13	1.54	0.191
R200V5	1.37	0.12	1.57	0.191
R200V6	1.36	0.13	1.55	0.191
Braza	1.38	0.12	1.60	0.190
R250	1.36	0.21	1.81	0.198
R400	1.36	0.22	2.26	0.206
R500	1.36	0.32	2.60	0.205

In [6], for Re<200, good agreement was found with other published results, which the present investigation also confirmed but the results of [6] were not so good at Re=200. The parameters of R200V1 correspond to those used in [6] and the present work suggests that the convection timestep is too large. This results in too large an average Cd and too small values of the ranges. When DTT<0.1, however, variation of the parameters subject to the earlier described limitations does not seem to cause significant changes in the results. Tables 1 and 2 also show typical values of parameters and results for Re up to 500. Figures 1 and 2 show the lift and drag versus time and streamlines at t=40 for R200V5 and R500 respectively.

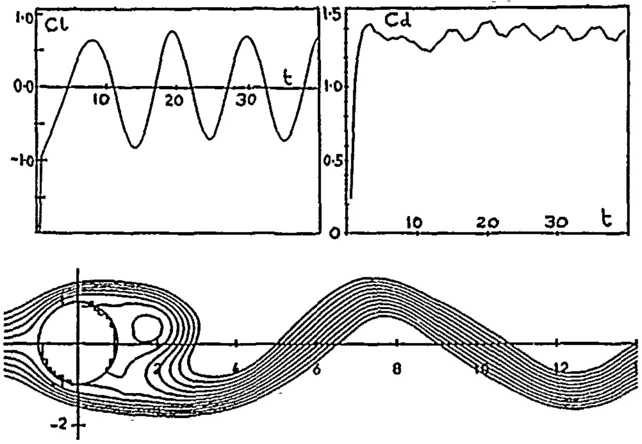


Fig.1. Lift, drag and streamlines at Re=200.

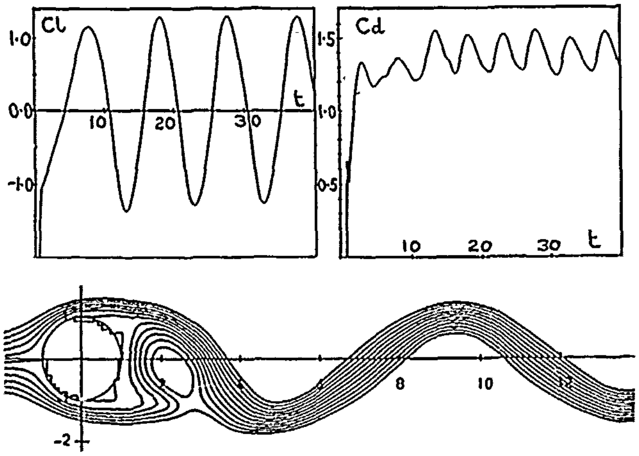


Fig.2. Lift, drag and streamlines at Re=500.

REFERENCES

1. Phil.Trans.Roy.Soc.A., 261,137-162 (1967).
2. ASME J.Basic Eng. 90,511-520 (1968).
3. ASME J.Fluids Eng. 111,5-52 (1989).
4. J.Fluid Mech. 57,785-796 (1973).
5. J.Fluid Mech. 194,45-77 (1988).
6. J.Fluids & Structures, 3,439-479 (1989).
7. Ph.D.Thesis, Manchester University (1990).
8. J.Comp.Phys. 9,440-465 (1972)
9. Ph.D.Thesis, Manchester University (1980).
10. J.Fluid Mech. 165,79-130 (1986).

NUMERICAL MODELING SEPARATION OF FLOW OF VISCOUS FLUID IN THE PIPES

OSTAPENKO V.A.
Faculty of Mathematics and Mechanics
State University of Dnepropetrovsk
pr. Gagarina 72
Dnepropetrovsk-10, 320625, U S S R

Abstract-It is proposed method for discavering separation regions of viscous fluid in bent pipes.That method is based on decomposition of full system equations of hydrodynamics of viscous fluid and consits in three stages. On the first stage the equations of streamlines are obtained and time- and space-average length streamline for given pipeline is calculated. Stagnating zones and reverse motions are obtained as well. On the second stage we construct one-dimensional equivalent model of viscous fluid motion along streamlines which takes into consideration the narrowing of area of transversal section by stagnating zones and reverse motions.Solving the initial-boundary value problem for equations one-dimensional equivalent model we obtain the subdomains of the pipeline at which are valid the necessary conditions for separation of flow.

The aim of third stage is the control of sufficient conditions for separation of flow. For that purpose the system equations of hydrodynamics for boundary layer of bent pipes the transversal section of which is near to circle is derived. For that system we solve initial-boundary value problem only in those subdomains of the pipeline at which are valid the necessary conditions for separation of flow and thus check sufficient conditions for separation of flow.

The application of described method permits to construct the pipelines with small energy losses because it is possible to prevent separation of flow.The decomposition of separation of flow problem decreases considerably the expenditures of computing time.

I. INTRODUCTION

Under some circumstances of the motion of fluid in the pipes can occur separation of flow and that phenomenon leads to sharp change of pressure, speed and temperature of fluid compared with their values without separation. Most essential cause which leads to separation of flow is the form of channel especially sharp change of its transversal section [1].

The separation of flow leads to essential increase of the resistance of fluid motion in the pipes. Therefore constructing pipelines with small energy losses it is necessary to create such form of channel that separation of flow should not occur.

To solve the problem of creatign pipelines in which the motion of fluid occurs without separation of flow it is not necessary at full measure to elaborate mathematical model of separation of flow. In that case is necessary only to answer the question: occurs or does not occur separation of flow in given pipeline under given parameters of fluid which flows in pipe. Last problem is essentially simpler and its solution can be obtaind by methods of hydrodynamics. At present paper is suggeted the method for solution such problem.

From the point of view hydrodynamics for arising separation of flow two factors are necessary: positive gradient of pressure and viscosity of fluid [1].It is known that in three-dimensional space separation of flow arises in boundary layer [1,2]. Exactly, the separation of flow occurs when nearest to wall of channel the streamline cames off wall. Therefore the necessary condition for separation of flow is

$$\left.\frac{\partial q}{\partial n}\right|_S = 0 \quad , \tag{1}$$

where S is contour of pipeline, $\bar{n}$ -unit vector of external normal to surface S, q-normal to S component of fluid velocity. The necessary condition (1) can lead to separation of flow if in the same region pressure p increases towards the flow, i.e. is valid condition

$$\left.\frac{\partial p}{\partial x}\right|_S > 0 \quad . \tag{2}$$

The sufficient condition for separation of flow is

$$\left.\frac{\partial u}{\partial n}\right|_S \leqslant 0 \quad , \tag{3}$$

where u is longitudinal component of fluid velocity.

From (1)-(3) it can be seen that to find the sections of pipeline in which are valid the necessary and sufficient conditions for separation of flow it is necessary to know the velocity and pressure fields in the pipe. To obtain those fields we have to solve the initial-boundary value problem for the full system of hydrodynamic differential equations [3].However obtaining exact solutions of such problems is impossible. Numerical integration of those problems even for simple regions demands so much computing time that the obtained results become insufficiently reliable in consiquence of the accumulation of errors of calculations. This situation becomes more complicated if it is necessary to optimize system parameters because it demands multiple computations for initial-boundary value problems.

So to solve separation of flow problem we propose a method of decomposition which consists in three stages. Main idea of the first two stages is to represent outside to boundary layer flow as one-dimensional one along streamlines and with the help of such conception to find the parameters of outside flow. For that purpose we construct one-dimensional equivalent model of the motion of viscous fluid in pipe [4].

II. FIRST STAGE

It is known that equations of motion of perfect fluid are one-dimesional towards streamlines. In case of viscous fluid the equations of motion are one-dimensional towards streamlines accurate to term which takes into consideration the rotation of velocity field [4]. The full system equations of hydrodynamics of viscous fluid is singularly perturbed one and if $\lambda=\mu=0$, where λ and μ are constants of Lame, becomes full system equations of hydrodynamics of perfect fluid. It is known,in the theory of singularly perturbed boundary value problems, the differens between solutions of perturbed and non-perturbed problems has order O(λ) or O(μ) everywhere with the exeption of boundary layer which thickness has the same order.

On the first stage of decomposition we find equations of streamlines for three-dimensional motion of fluid in pipe. Above considerations permit to find those equations by means of solving initial-boundary value problem for perfect fluid. After the velocity field for perfect fluid is obtained we average that one for time of activity of pipline, calculate length of every average streamline and calculate average length of streamlines in pipeline.

Ш. SECOND STAGE

On the second stage there we construct one-dimensional equivalent model of viscous fluid motion in the pipeline. The length of one-dimensional region of motion is equal to average length of streamlines. The main task for creating the one-dimensional model is to take into consideration the influence of the forces of friction on the motion of fluid in pipeline. That model is not ordinary consequence of three-dimensional motion of viscous fluid under only assumption that all functions which are inserted into equations depend on one space variable but is more complite one. It takes into account, in particular, the change of the area of transversal section and perimeter of pipelines towards the flow (along x-axis) and the heat exchange with the walls of pipeline and is described by following equations [4]

$$\frac{\partial}{\partial t}(\rho VS)+\frac{\partial}{\partial x}(\rho V^2S)+\frac{\partial}{\partial x}(pS)+\frac{\nu\rho}{2}V^2Q=0$$
$$\frac{\partial}{\partial t}(\rho S)+\frac{\partial}{\partial x}(\rho VS)=0$$
$$\frac{\partial}{\partial t}[\rho S(c_vT+V^2/2)]+\frac{\partial}{\partial x}[\rho VS(c_vT+V^2/2)]+\frac{\partial}{\partial x}(pVS)+$$
$$+\nu\rho V^3Q/2+k(T-T_w)Q=0$$
$$p=\rho TR/\gamma \quad . \tag{4}$$

It should be noted that in one-dimensional model there are taken into consideration additionally the results of solving of three-dimensional problem on the first stage.By means of analysis of velocity field in the pipeline, in particular, we discover stagnating zones and reverse motions and using those results make correction of geometrical values of area of transversal section and perimeter of the boundary contour of pipeline.

As the result of solving initial-boundary value problem for system (4) we obtain the values which characterise motion of viscous fluid in pipeline. In particular, with the help of function of pressure p(x,t) we can discover sections of pipeline in which condition (2) is valid, i.e. such sections where separation of flow may occur in principle.

IV. THIRD STAGE

To answer the question whether separation of flow really occurs we have to solve three-dimensional initial-boundary value problem for full system of hydrodynamic differential equations. However, taking into consideration that separation of flow occurs only in boundary layer, it is sufficient to consider that problem only there,moreover only in those parts of boundary layer where condition (2) is valid. In its turn the small thickness of boundary layer permits to simplify equations of motion in it.

Let us consider equations of hydrodynamics of viscous fluid in cylindrical channel with circular cross-section of radius R_1. Let us introduce cilindrical polar coordinates r,φ,x and transformation $\xi=R_1-r$.We introduce as well the transformation to undimensional variables (with asterisk) with the help of scaling

$$x=l_0x^*;\quad \xi=\delta_0\xi^*;\quad \varphi=\varphi_0\varphi^*;\quad V_x=V_x^0V_x^*;\quad V_\varphi=V_\varphi^0V_\varphi^*;$$
$$V_z=V_z^0V_z^*;\quad \mu=\mu_0\mu^*;\quad p=p_0p^*;\quad \rho=\rho_0\rho^*;\quad t=t_0t^*;$$
$$k=k_0k^*;\quad T=T_0T^* \tag{5}$$

and conditions of connection

$$t_0=l_0/V_x^0;\quad p_0=\rho_0V_x^{0^2};\quad \nu_0=\mu_0/\rho_0;\quad l_0/\delta_0\cdot V_z^0/V_x^0=1$$
$$V_\varphi^0/\varphi_0=V_x^0/l_0=V_z^0/\delta_0;\quad \nu_0=\delta_0V_z^0;\quad k_0=R\mu_0/\gamma. \tag{6}$$

Executing in full system of hydrodynamic differential equations all of these transformations, evaluating every term and neglecting the terms which are sufficiently small compared with unity, we derive system equations for boundary layer (again in dimensional variables)

$$\frac{1}{\rho}\frac{\partial p}{\partial\xi}=0$$
$$\frac{\partial V_\varphi}{\partial t}-V_z\frac{\partial V_\varphi}{\partial\xi}+\frac{V_\varphi}{R_1}\frac{\partial V_\varphi}{\partial\varphi}+V_x\frac{\partial V_\varphi}{\partial x}=-\frac{1}{R_1\rho}\frac{dp}{d\varphi}+\frac{\mu}{\rho}\frac{\partial^2V_\varphi}{\partial\xi^2}$$
$$\frac{\partial V_x}{\partial t}-V_z\frac{\partial V_x}{\partial\xi}+\frac{V_\varphi}{R_1}\frac{\partial V_x}{\partial\varphi}+V_x\frac{\partial V_x}{\partial x}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{\mu}{\rho}\frac{\partial^2V_x}{\partial\xi^2}$$
$$\frac{\partial\rho}{\partial t}-\frac{\partial(\rho V_z)}{\partial\xi}+\frac{1}{R_1}\frac{\partial(\rho V_\varphi)}{\partial\varphi}+\frac{\partial(\rho V_x)}{\partial x}=0$$
$$\rho c_p\left(\frac{\partial T}{\partial t}-V_z\frac{\partial T}{\partial\xi}+\frac{V_\varphi}{R_1}\frac{\partial T}{\partial\varphi}+V_x\frac{\partial T}{\partial x}\right)-\left(\frac{\partial p}{\partial t}-V_z\frac{\partial p}{\partial\xi}+\frac{V_\varphi}{R_1}\frac{\partial p}{\partial\varphi}+\right.$$
$$\left.+V_x\frac{\partial p}{\partial x}\right)=\mu\left[\left(\frac{\partial V_\varphi}{\partial\xi}\right)^2+\left(\frac{\partial V_x}{\partial\xi}\right)^2\right]+k\frac{\partial^2T}{\partial\xi^2} \quad . \tag{7}$$

Now for the solution of separation of flow problem we have to solve initial-boundary value problem for system (7) in boundary layer. The motion of fluid oitside boundary layer with the high degree of accyracy can be described by parameters of one-dimensional equivalent flow. After solving such problem we can verify the satisfaction of conditions (1) and (3). If those conditions are valid it means that form of pipeline have to be changed and for new form we have to solve again the same problem.

This technique can be applied as well for bent pipelines with near to circle cross-section.To take into consideration of centrifugal forces the first equation (7) has to be written as

$$\frac{\partial p}{\partial\xi}=-\frac{\rho V_x^2\cos\varphi}{R(x)+R_1(1-\cos\varphi)}$$

and to right-hand side of second equation (7) must be added the term

$$-\frac{V_x^2\sin\varphi}{R(x)+R_1(1-\cos\varphi)} \quad ,$$

where

$$R_1=\frac{1}{2\pi}\int_0^{2\pi}z(\varphi)d\varphi \quad .$$

The method which has been proposed permits to create packaged programs for analysis separation of flow in the pipes. Using that programs we can design pipelines with small losses of energy because of that phenomenon there do not occur.

REFERENCES

1.Chang P.K.Separation of Flow,Pergamon Press, Oxford,1970.

2.Chang P.K. Control of Flow Separation, Hemisphere Publishing Corporation,Washington,1976.

3.Ostapenko V.A.On the putting of initial-boundary value problems for motion of gas in pipelines of internal-combustion engines,"Hydrodynamics and Theory of Elasticity", Dnepropetrovsk,DSU,42-47,1987 (in Russian).

4.Ostapenko V.A.Equivalent model of motion of viscous gas,"Differential equations and their Applications to Physics",Dnepropetrovsk,DSU,4-13,1990 (in Russian).

COMPUTATIONAL ANALYSIS OF THREE-DIMENSIONAL SHOCK-WAVE/TURBULENT BOUNDARY LAYER INTERACTION

A. KOURTA and H. HA MINH

Institut de Mécanique des Fluides de Toulouse
I.N.P.T.- U.R.A. C.N.R.S. 0005
Av. du Prof. Camille Soula, 31400 Toulouse-Cedex FRANCE

and

Centre Européen de Recherche et de Formation Avancée
en Calcul Scientifique (C.E.R.F.A.C.S.)
42, avenue G. Coriolis, 31057 Toulouse-Cedex FRANCE

ABSTRACT :

The topic of this paper is the prediction of three-dimensional flows and the analysis of shock wave/turbulent boundary layer interaction. For these, a two-step method based on the MacCormack scheme has been used. The averaged Navier-Stokes equations with turbulence model (Baldwin-Lomax) were solved numerically in a general coordinate system for three dimensional turbulent flows. The numerical computation was performed for a 3D channel flow. The configuration is the same as the one studied experimentally at ONERA [1], [2] in a wind tunnel. It consists of a bump mounted in the lower wall. The countoured portion of the bump is at an angle of 60^0 with the upstream flow. The flow is then highly three-dimensional. The features of this flow are predicted. A detailed description of a 3D interaction and a complete analysis of the physical phenomena and the comparison with the experimental results are done.

I. INTRODUCTION

Several Navier-Stokes codes have been developped in order to calculate two or three dimensional viscous compressible flows. the three dimensional code is needed when we have to analyze complex fluid flows. The study of three dimensional flow has become a subject of significant importance in the aerodynamic.

The purpose of this investigation is to develop a code based on a MacCormack method for solving the compressible Navier-Stokes equations to solve three dimensional turbulent flows.

The aim of this study is to test the capability of this code to simulate 3D shock wave/turbulent boundary layer interaction. This phenomena is one of the interest gaz dynamic problems. The effects of such interaction are the increase of the aerodynamic drag and the possibility of buffet onset. The natural consequence of this interaction is the separation. Hence, experimental investigations[1,2,3] were done to provide a set of data and to analyse this interaction.

In our case, the numerical computation was performed for a 3D transonic channel flow. The configuration is the same as one studied experimentally by BENAY et al [1] in a wind tunnel. The flow is highly three dimensional with the strong shock wave/boundary layer interaction.

II. GOVERNING EQUATIONS

If we consider the case of homogeneous flow excluding chemical reactions or very high temperature effects, the full Navier-Stokes equations constitute a good physical model. This model describes conservation of mass, momentum and total energy. With the mass-weighted averaged Navier-Stokes equations, a turbulence model is used to relate the turbulent flux terms to the mean flow parameters, through the use of an eddy viscosity coefficient. The turbulent viscosity is obtained by the Baldwin-Lomax model [4].

III. NUMERICAL METHOD

The numerical method is an explicit version of a MacCormack scheme [5, 6]. The predictor and corrector structure is used. For each time step, an explicit increment is evaluated by using the forward or backward approximations for the inviscid part and the central difference for the viscous terms.The scheme is a finite volume, cell centre method.

In order to improve numerical efficiency, the second order accurate flux splitting has been included in the method. The flux splitting was motivated by one need for a better description of discontinuities and a more rigourous treatment of the boundary conditions. The flux splitting used here is close to the one developed by Steger and Warming [6, 7].

IV. RESULTS

The numerical method, described above, has been applied to the calculation of the transonic flow in a three dimensional channel. The forth boundaries are treated as a no slip boundaries. Given the total pressure and total temperature, the inlet flow angle and the outlet downstream static pressure, calculations are performed. The same conditions used in the experimental study were implemented. The total pressure is 92 kPa, the total temperature is 300 K and the Reynolds number at the sonic state with the throat width as length scale is $1.13 * 10^6$.

The features of this flow are predicted. Figures (1) and (2) show iso-Mach lines in the longitudinal vertical planes, at two different sections. These plots reveal the existence of the lambda shock comprising of a front oblique shock and a quasi normal rear shock. The second foot of the lambda shock is not very intense as it is in a two dimensional configuration. The system of the shocks is generally associated with a strong interaction entailing separation. The length of the separation zone in this case is smaller than in a two diemensional configuration. The results are in good agreement with the experimental data. The main difference with a 2D shock-wave/boundary-layer interaction is that in a 3D situation, the flow can take a transverse or crosswize direction, and consequently when separation occurs we generally do not observe recirculating bubble in the streamwise direction.

Isobar lines on the bump are shown in figure (3). It reveals a nearly two dimensional incoming flow which is turned by a shock forming slightly downstream of the top of the bump. In this figure the shock foot is observed. At the shock foot, interaction with the boundary layer takes place entailing separation. The three dimensional Character of the flow can also be observed downstream of the top of the bump.

Comparison to experimental data is given in figure (4). In this figure we plot the Mach number profile in one section before the shock. The agreement seems to be good in the external inviscid flow. In the near wall regions, the calculation overestimates the thickness of the boundary layers. This is due to the overestimation of the turbulence viscosity and perhaps to the coarse mesh used.

V. CONCLUSIONS

A three dimensional code based on a Mac Cormack scheme has been developed. It has been applied to the calculation of transonic turbulent channel flow. The features of this three dimensional flow were predicted and the shock-wave/boundary-layer interaction was analysed. Due to the calculation of turbulent viscosity, the thickness of the boundary layers is larger than the experimental one.

Acknowledgments

The authors would like to acknowledge helpful discussions with Prof. D. Vandromme. Thanks also to T. Pot from ONERA for providing technical reports.

References

[1] BENAY R., POT T., Experimental study of shock-wave boundary-layer interaction in a three-dimensional channel flow, Symposium IUTAM, Ecole Polytechnique Palaiseau, 9-12 sept 1985.

[2] BENAY R., POT T., DELERY J., Etudes fondamentales sur les interactions onde de choc-couches limite dans un canal tridimenstionnel, AGARD/PEP, Munich, RFA, 10-12 sept. 1986.

[3] DOERFFER P., DALLMANN U., Reynolds number effect on separation structures at normal shock wave/turbulent boundary-layer interaction, AIAA Journal, vol. 27, n° 9, pp. 1206-1212, September 1989.

[4] BALDWIN B.S., LOMAX H., Thin layer approximation and algebraic model for separated turbulent flows, AIAA paper n° 78-257, 1978.

[5] MAC-CORMACK R.W., Current status of numerical solutions of the Navier-stokes equations, AIAA Paper 85-0032, 1985.

[6] MAC-CORMACK R.W. A numerical method for solving the equations of compressible viscous flow, AIAA paper 81-0110, 1981.

[7] STEGER J.L., WARMING R.F., Flux vector splitting of the inviscid gas dynamics equations with application to finite difference method, J. Comp. Phys. vol. 40, n° 2, p. 283, 1982.

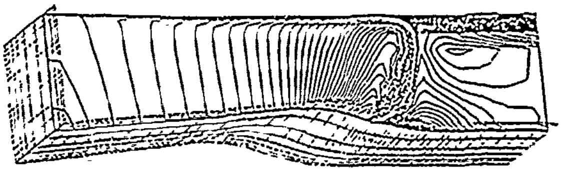


Figure (1): Mach number contours Y=27.mm

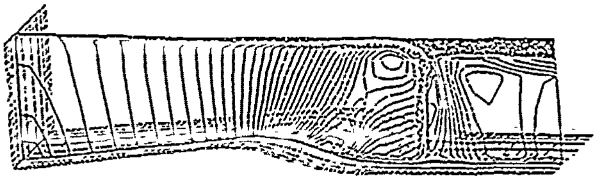


Figure (2). Mach number contours Y=89.mm

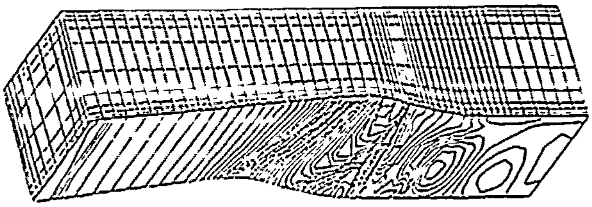


Figure (3): Pressure contours at the lower wall

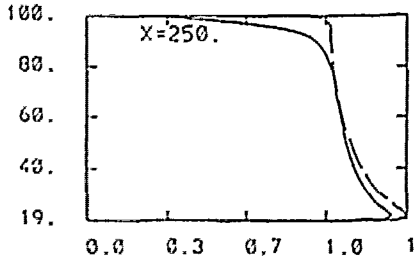


Figure (4): Mach number profile Y=27.mm
– – exp.

The Theory and Computation of th϶ Second-order Water Wave Forces upon a 2-D Floatıng Body

Tang Ling
Shanghai Institute of Applied Mathematics & Mechanics
Shanghai University of Technology,Yan Chang Road,Shanghai 200072,PRC

Abstract:In this paper,the expressions of the second o order water wave forces upon a 2-D floating body are derived with the perturbation method,and the numerical computations are carried out with the boundary element method at the same time. Compared with the results of the experiments and of the calculations done by other researchers, the present results are more resonable and accurate.

I. Introduction

These years, with the rapid development of the ocean engineering,much attention was paid to the nonlinear water wave theory. This is because that the water wave forces are the most important loading to many marine structures such as ships and drug platform as well as some offshore structures. It is known that a body will be subjected to linearand nonlinear force. In regular waves,we call the latter second-order wave forces which include two parts,the steady drifting forces with frequency equal to zero and the biharmonic oscillating ones with twice of the incident wave frequency. Both are veryimportant factors and need fully consideration.

Early in 1960,Maruo,H[1] obtained a general expression for the horizental steady drifting force of a 2-D body,which is proportional to the square of the reflected wave amplitude of the linear solution. His work was generalized later by Newman[2] and Faltinsen [3], It is reported that there is a good agreement between the results of experiments and of calculations with this method. But it cannot predict the biharmonic wave force which also covers a considerate proportion in the total second-order wave forces.

In 1980s, a more direct investigation was carried out by Kyozuka[4]. With the boundary element method, he had obtained the solution of the diffractional and radiational problems in regular waves. Recently,the problem of a fixed 2-D body in regular waves was studied by Liu & Miao[5],they introduced the Dirac function to the inhomogenous free surface condition and found a complete and consistent solution for the second-order diffraction potential.

In this paper, the problem of a free body floating in regular waves is investigated. It is an pofound research which is an extension of the previous work. The problems are solved respectively,which include the second-orderdiffraction of incident waves and the coupled effects of the first order motions of the floating body on second-order wave forces.

II. The Governing Equation and Boundary Conditions

As depicted in Fig.1,the two Cartesian coordinate systems are related each other by the following expressions:

$$X=S1+\underline{X}\cos(S3)-\underline{Y}\sin(S3)$$
$$Y=S2+\underline{X}\sin(S3)-\underline{Y}\cos(S3)$$

where S1,S2,S3 are,respectively, th϶ excursions in X and Y directions and rotation of the body about the origin $\underline{O}$.

We assumed that the fluid is incompressible and invicid, the flow irrotational; thus the motion of the fluid can be described by a velocity potential H,and the governing equation is the Laplace equation:

$$\nabla^2 H(x,y,t)=0$$

Since the floating body is impermeable, we have

$$Hn=Vn,\text{on the surface of the body}$$

Through analysis, we obtain the general free surface condition

$$H_t+gHy+2HxHx_t+2HyHy_t+\dot{H}xHxx+\dot{H}yHyy+2HxyHxHy = 0$$

At the bottom of infinity, we have

$$\text{Lim } Hy = 0$$

Besides, a radiation condition should be imposed at infinity,i.e. the diffraction waves must be propogating outward from the body.

III. Perturbation Analysis

For infinite deep water, the potential of incident wave H_I can be expressed as

$$H_I= ga/(iw)*\exp(ky+ikx-iwt)$$

in which w,k,a are wave frequency,wave number and wave amplitude respectively.

Let e=ka be the small wave slope, by means of perturbation technique, the potential H has the form

$$H(x,y,t)=Re[e\underline{Hs}\exp(-iwt)+e*e\underline{Hc}\exp(-i2wt)]$$

Since the amplitude of Si(i=1,2,3) are also very small,so we have

$$\sin(S3)=S3+ O(e*e*e)$$
$$\cos(S3)=1-0.5S3*S3+ O(e*e*e)$$

Thus,

$$X = S1+\underline{X}-\underline{Y}S3-\underline{X}S3*S3/2+O(e*e*e)$$
$$Y = S2+\underline{Y}+\underline{X}S3-\underline{Y}S3*S3/2+O(e*e*e)$$

Similarly

$$n_1 = \underline{n}_1-\underline{n}_2S3-\underline{n}_1S3*S3/2$$
$$n_2 = \underline{n}_2+\underline{n}_1S3-\underline{n}_2S3*S3/2$$

Since $Vn=\dot{S}1\underline{n}_1+\dot{S}2\underline{n}_2+\dot{S}3(\underline{x}\underline{n}_2-\underline{y}\underline{n}_1)$ (where the overdot denotes the derivation about the time t.),Let the instantaneously wetted contour of the body be C(x,y,t) and in still water it takes the form of Co($\underline{x}$,$\underline{y}$). Following the same procedure, the free surface condition can be written as

$$[Hn]c=(Hxn_1+Hyn_2)c\ ,\text{i.e.,}$$
$$(\overset{(1)}{H}n)c=\dot{Q}i\ ni$$
$$(\overset{(2)}{H}n)c=Q3[\overset{(1)}{H}x(\underline{x},\underline{y})\underline{n}_2-\overset{(1)}{H}y\ \underline{n}_1)-(Q1-Q3\ \underline{Y})[\overset{(1)}{H}xx\ \underline{n}1+\overset{(1)}{H}xy\ \underline{n}2]$$
$$-(Q2+Q3\ \underline{X})[\overset{(1)}{H}xy\ \underline{n}1+\overset{(1)}{H}yy\ \underline{n}2]=f(\underline{x},\underline{y})$$

where Si=e Qi(x.y) exp(-iwt)

Let $\overset{(1)}{H}=H_I+Qi*Ji$, (i=1,2,3),in addition it must include first order diffractional potential $\overset{(1)}{H}d$.

The equations of different orders are

1st: $\nabla^2\overset{(1)}{H}d=0$,in fluid domain

$$\overset{(1)}{H}dy-k\overset{(1)}{H}d=0,\ \text{at } y=0$$
$$\overset{(1)}{H}dn=-H_In,\ \text{on } Co(\underline{x},\underline{y})$$
$$\text{Lim } \overset{(1)}{H}dy=0$$
$$\lim (Hdx\underline{+}ikHd)=0$$

and

$$\nabla^2 Ji=0,\ \text{in fluid domain}$$

Jiy-ikJi=0, at y=0

Jin=ni, on Co(x,y)

Lim Jiy = 0

Lim (Jix ± i k Ji) =0

The first order problem can be solved with the Frank method numerically.

2nd order:

∇^2Hl =0, in fluid domain

Hly - 4k Hl = 0, at y=0

[Hln]c =f(x,y)-H2n-H3n, on Co(x,y)

Lim Hly = 0

Lim (Hlx ± i 4k Hl) = 0

∇^2H2 = 0, in fluid domain
H2y - 4kH2 = Pl(x), at y=0
Lim H2y = 0, y approaches infinity
Outgoing waves at infinity

∇^2H3 = 0 , in fluid domain
H3y - 4kH3 = P2(x), at y=0
lim H3y = 0, y approaches infinity
Outgoing waves at infinity

where P=(iw/2g)[2($Hx^{(1)2}$+$Hy^{(1)2}$)-$H^{(1)}$($Hyy^{(1)}$-k$Hy^{(1)}$)] , and

Pl= -4ik*ka∫c[$Hn^{(1)}$ -$H^{(1)}$d/dn]exp(ky+ikx)·dl, x≤-B/2

pl= 0, x> -B/2

and

p2=p-pl

It is clear that Hl will be found out with the same process used in 1st order once the potential H2 and H3 has been solved. Here are the solutions of H2 and H3 (The process was omitted for brevity)

H2 = -sgn(x+B/2) (Pl/4k)exp(4ky+i4k x+B/2)-rPl/4k
+sgn(x+B/2) (Pl/3.1416)

$\int_0^\infty$[mcos(my)+4ksin(my)]/(m +16mk) exp(-mx+B/2)·dm

where sgn(x+B/2)=-1,r=1, when x<-B/2
sgn(x+B/2)=1, r=0, when x>-B/2

H2 approaches -Pl/8/3.14156 when x approaches 0.

H3 = Re[-(1/3.1416)$\int_{-\infty}^{\infty}$P2(s)·ds PV.$\int_0^\infty$exp(-jvz+jvs)
/(v-4k)·dv -i$\int_{\infty}^{\infty}$P2(s)exp(-4jkz+4jks)·ds]

where z=x+jy and PV. denotes Cauchy principle integrates.

IV. Equation of 1st order Motions

According to theorem of Newton, we have

$$M(\ddot{S}1 - \underline{Y}g\,\ddot{S}3) = e\,F1$$

$$M\,\ddot{S}2 = e\,F2 + pgA - Mg$$

$$\underline{I}\,\ddot{S}3 - Mg\,\underline{Y}g\,\ddot{S}3 + M(gS1 - \underline{Y}g\,\ddot{S}1) = e\,F3$$

where M and Yg are mass and coordinate of the center of gravity in Yaxis and I=∫c.(x +y)dM of the body.

In still watwr,the body is in equilibrium,that means M g =pgA. where p is the density of fluid. By using pressure equation we finally obtained

$$\begin{bmatrix} -w\,(M+D11)-iwG11, & 0, & -w\,(D31-M\underline{Y}g)-iwG31 \\ 0, & -w\,(M+D22)+pgB-iwG22, & 0 \\ -w\,(D13-M\underline{Y}g)-iwG13, & 0, & -w\,(\underline{I}+D33)-iwG33+pg\underline{Iw}-pg(\underline{Y}b+\underline{Y}g) \end{bmatrix}$$

$*[Q1,Q2,Q3]^T = p[T1,T2,T3]^T$

where Dij+(i/w)Gij=p∫cJinjdl ,(i,j=1,2,3)
Tj=∫c.(H_I+H_d)(iw)njdl,Dij and Gij are called the added masses and coefficients of damping.

AYD=∫c.x ydy, A is the submergerd area of the body and Iw is the moment of inertial of wetted section about the x axis.

Since F =Re[e $F^{(1)}$exp(-iwt)+e*e($F^{(0)}$+$F^{(2)}$exp(-i2wt)] and F = -p∫c (H +gy)nidl, so the expressions of forces with different order will be derived easily and is not listed here,readers may be referred to [7].

V. Results and Conclusion

As depicted in Fig.2, B=2.0(m), h=0.8(m), A=0.76 Yg=-0.031,Iw=0.667,I=108.30,k=1.0(1/m),M=77.52(Kg/m) p=102,g =9.81,Yb=0.829.

The computation results are showed in Fig.3 and Fig.4. It was carried out on the vax-11/750 digital computer in Shanghai.

From these computation,we may clearly found out that the present results seem agree rather well with the experimental ones. Thus,we may conclude that

1. The magnitude of biharmonic forces have the same order compared with the steady drifting forces,further more, they reveal the oscillating nature of high order forces.
2. The presen theory can be used to predict the wave forces in irregular waves combined with Fourier analysis and application may also be found in 3-D calculation by means of method of stripping in seakeeping theory.

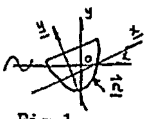


Fig.1

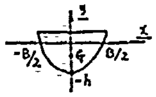


Fig.2

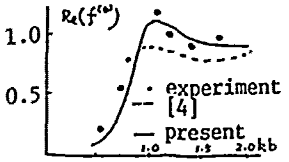


Fig.3 2nd order drifting force(Steady)

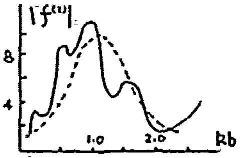


Fig.4 2nd order heaving force(bihar.)

Reference

1. H,Maruo, The Drift of a Body Floating on waves, J. Ship Res.,Vol.4, No.3, pp.1-10, 1960

2. J.N. Newman, The Drift Force & Moment on Ships in Waves, J. Ship Res., Vol.11, No.3, pp.20-29,1967

3. Faltisen,O.M.& F,Michelsen, Motion of Large Structures in Waves at Zero Froude Number, Symp. of the Dynamics of Marine Vehicles & Structures in Waves, London,1974.

4. Y,Kyozuka, Experimental study on 2nd-order Forces Acting on a Cylindrical Body in Waves, 14th Symp. on Naval Hydrodynamics,Michigen,U.S.A.,Aug.23-27, Session III & IV, Preprints pp.73-136.

5. Liu Y.Z. & Miao G.P., 2nd-order Water Wave Forces on a 2-D Body, Chinese Ship Building,No.3,pp1-14, 1985.

6. Miao G.P., Second-order Wave Forces on a Cylinder of Large Diameter, Chinese Ship Building, No.3, pp.12-34,1987

7. Tang L.,Miao G.P. & Liu Y.Z., Second-order Water Wave Forces on a Floating Object, J. of Shanghai University of Technology,Vol.11, No.6,1990.

8. Lamb H., Hydrodynamics,6th Ed.,Cambridge Press, 1932.

ON PERMANENT CAPILLAR-HEAVY WAVES IN FINITE CHANNELS

NABIL MOUSSA
Associate Professor and Mathematics Unit Head
The American University in Cairo, P.O. Box 2511, Cairo, EGYPT

Abstract-This work is based on the constructive existence proof of solutions of a comprehensive class of nonlinear free boundary value problems of plane hydrodynamics by E. Zeidler [5] and a general computational method for constructing the solutions numerically given by the author [3]. The case of permanent capillar-heavy waves in finite channels will be considered.

INTRODUCTION AND NOTATIONS

After formulating the boundary value problem in the stream plane in section 1, we apply Levi-Civita conformal mapping to get a nonlinear boundary value problem on the circular ring and transform the problem to an operator equation in the Banach space C_ψ in section 2. In section 3 we formulate the statement of existence and uniqueness. The general computational method for constructing the solutions will be used in section 4 and in section 5 the solutions in different orders will be computed.

For the function $f(\rho,\sigma)$ defined in the open circular ring $E_{q1}=\{\zeta:q<|\zeta|<1:\zeta=\rho e^{i\sigma}\}$, the symmetry behaviour will be denoted by

$$f\epsilon C: f(\rho,-\sigma) = f(\rho,\sigma); \quad f\epsilon U: f(\rho,-\sigma) = -f(\rho,\sigma)$$

with $f(1,\sigma) \equiv f(\sigma)$.

For the Fourier series of $f(\sigma)$

$$f = a_o + \sum_{k=1}^{\infty} (a_k \cos k\sigma + b_k \sin k\sigma)$$

we have the closure condition

$$a_o(f) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f \, d\sigma = 0$$

Further we define the following operators (see [1]):

$$Tf = f - a_o(f) \; ; \quad Jf = T \int_0^{\sigma} Tf \, d\sigma \; ;$$

$$Kf = \frac{1}{2\pi} \int_{-\pi}^{\pi} \cot \frac{\sigma' - \sigma}{2} (f(\sigma')-f(\sigma))d\sigma' \; ;$$

$$V_q f = \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{q^k}{1 - q^{2k}} \int_{-\pi}^{\pi} \sin k(\sigma - \sigma')f(\sigma')d\sigma' \; ;$$

$$W_q f = - \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{q^{2k}}{1 - q^{2k}} \int_{-\pi}^{\pi} \sin k(\sigma - \sigma')f(\sigma')d\sigma' \; ;$$

$$X_q f = Kf + W_q f \quad ; \quad 0 \leq q < 1$$

and the following Stoke's parameters

$$s_k = q^{-k} + q^k \quad ; \quad d_k = q^{-k} - q^k$$

$$t_k(q) = \frac{s_k}{d_k} = \frac{1 + q^{2k}}{1 - q^{2k}} \; ; \quad \pi_k(q) = k \, t_k(q)$$

$0_r(\theta,\tau)$ will denote regular power series with power greater or equal to r in θ.

1. Problem Formulation

To describe permanent capillar-heavy waves of an ideal homogeneous incompressible liquid in a finite channel, we follow the method of Levi-Civita [2] by considering the two systems of coordinates:

(i) The moving x-y system in which the surface seems to rest;

(ii) The X-Y rest system in which the liquid is resting on the ground (see Figure 1).

We have $x=X+ct$, $y=Y$ with c the speed of propagation of the wave in the sense of Levi-Civita. The free surface $y_0(x)$ satisfies

$$(1) \quad y_0(-x) = y_0(x); \quad y_0(x+\lambda) = y_0(x);$$

$$(2) \quad \int_{-\lambda/2}^{\lambda/2} y_0(x)dx = 0$$

The flow potential $W(z)=\phi+i\psi$ in x-y system will be normed through

$$(3) \quad W(iy_0(0)) = 0$$

As the free surface and the channel ground are flow lines, then

$$(4) \quad \psi(x,y_0(x)) = 0; \quad \psi(x,-h) = -\psi_0 < 0$$

with $\psi_0 > 0$ the flow in the moving system.

For the velocity $\bar{W}'=u+iv$ let us assume, as Levi-Civita proposed, that

$$(5) \quad W'(z) = \bar{c}\, e^{-iw} , \quad w = \theta + i\tau, \quad a_0(\tau(\rho,\sigma)) = 0$$

where

$$\bar{c} = c\, a_0(e^{-\tau(\rho,\sigma)}) = c(1+0_2(\tau))$$

$$(6) \quad w = i\, \ln(W'(z)/\bar{c})$$

$$\theta = \arg(u+iv), \quad \bar{c}\, e^{\tau} = \sqrt{u^2 + v^2}$$

Consequently, we have the following boundary value problem:

Find $y_0(x)$ satisfying (1) and (2) such that in the flow region $G=\{z:-h<y<y_0(x)\}$ there exists a holomorphic function $W(z) = \phi + i\psi$ satisfying (3) and (4). This function satisfies also

$$(7) \quad y = -h: \operatorname{Im} W'(z) = 0, \quad \text{i.e.,} \; \theta = 0$$

on the ground.

Along the free surface the Bernoulli's equation

$$(8) \quad \begin{aligned} y = y_0(x) &: g\mu y_0(x) + \tfrac{1}{2}\mu|W'(z)|^2 + p_i = K_0 \\ p_i &= p_0 - \beta \frac{d\theta}{ds} \end{aligned}$$

is satisfied, where μ is the density, β is the surface tension constant, p_i and p_0 are internal and external pressures respectively.

2. Conformal Mapping and Back Transformation

Due to periodicity of W(z) we consider the strip

$$G_z: -\frac{\lambda}{2} \leq x \leq \frac{\lambda}{2} , \quad -h \leq y \leq y_0(x)$$

It follows that

$$G_W: -\frac{c\lambda}{2} \leq \phi \leq \frac{c\lambda}{2} , \quad -\psi_0 \leq \psi \leq 0$$

is the conformal mapping using W(z).

Then we map the strip G_W using

$$\zeta = e^{i\sigma} = e^{-2\pi iW/c\lambda}$$

to the circular ring $q = e^{2\pi\psi_0/c\lambda} \leq |\zeta| \leq 1$ with a cut on the negative real axis (see Figure 1).

The back transformation will yield the free surface

$$(9) \quad \begin{aligned} x &= - \frac{\lambda}{2\pi}\, c/\bar{c} \int_0^{\sigma} e^{-\tau(\sigma)} \cos\theta(\sigma)d\sigma \\ y &= - \frac{\lambda}{2\pi}\, c/\bar{c} \int_0^{\sigma} e^{-\tau(\sigma)} \sin\theta(\sigma)d\sigma + y_0(0) \end{aligned}$$

where $y_0(0)$ is to be determined using (2).

The Bernoulli's equation (8) will be transformed in ζ-plane to

$$g\mu(y_0(0)-\frac{\lambda}{2\pi}\int_0^\sigma e^{-\tau(\sigma)}\sin\theta(\sigma)d\sigma)+\tfrac{1}{2}\mu c^2e^{2\tau(\sigma)}+p_i=K_0 \tag{10}$$

which is equivalent to

$$s_1(|z|=1):-\tilde{b}\,\frac{d\theta}{d\sigma}=\tfrac{1}{2}(e^{\tau}+ae^{-\tau})-\tilde{p}e^{-\tau}\int_0^\sigma e^{-\tau}\sin\theta\, d\sigma$$

$$s_q(|z|=q):\theta(q,\sigma)\equiv 0 \tag{11}$$

with

$$\tilde{p}=\frac{g\lambda}{2\pi c^2}.c^3/\bar{c}^3\ ,\ a=\frac{2(g\mu y_0(0)+p_0-K_0)}{\mu\,\bar{c}^2},\ \tilde{b}=\frac{2\pi\beta}{\lambda\mu c\bar{c}}$$

As $\theta\varepsilon U$, $\tau\varepsilon C$, we apply the operator T to the integrand and get

$$S_1:\ -\frac{d\theta}{d\sigma}=\frac{p}{2}(e^{\tau}+a'e^{-\tau})-be^{-\tau}\,Je^{-\tau}\sin\theta$$

$$S_q:\ \theta(q,\sigma)\equiv 0$$

(12) with

$$p=\frac{\lambda\mu c\bar{c}}{2\pi\beta}\ ,\ a'=a-2p\,a_0(\int_0^\sigma T\,e^{-\tau}\sin\theta\, d\sigma)$$

$$b=(g\lambda^2\mu/4\pi^2\beta)(c^2/\bar{c}^2)$$

The transformed boundary value problem will then be:

Find a function $w(\zeta)=\theta+i\tau\varepsilon C_\nu(G)$ with $\theta\varepsilon U$, $\tau\varepsilon C$ and the norm condition $a_0(\tau)=0$ which satisfies (12) on S_1 and S_q.

A necessary condition for the solution of this boundary value problem is

$$0=a_0\{\frac{p}{2}(e^{\tau}+a'e^{-\tau})-b\,e^{-\tau}\,J\,e^{-\tau}\sin\theta\}$$

which yields

$$a'=\frac{2b\,a_0(e^{-\tau}\,J\,e^{-\tau}\sin\theta)-p\,a_0(e^{\tau})}{p\,a_0(e^{-\tau})} \tag{13}$$

Replacing the constant a' in (12) by the functional $a'(\theta,\tau)$ and integrating using the operators T and J, we get the operator equation

$$\theta=pJX_q\theta+bJ^2\theta+JN(\theta,\tau=-X_q\theta) \tag{14}$$

where $N(\theta,\tau)$ is an expression of the form $O_2(\theta,\tau)$.

The linearized problem

$$\theta=pJX_q\theta+bJ^2\theta \tag{15}$$

has the solution

$$p=\bar{p}_{n,b,q}\equiv\frac{n^2+b}{\pi_n(q)}\ ,\ \theta=\bar{\theta}_n=\sin n\sigma,\ n=1,2,\ldots \tag{16}$$

3. Statement of Existence and Uniqueness

Given n,q,b with $0\le q<1$ and $b\neq\frac{\pi_n(q)-\pi_k(q)}{n^2-k^2}$ there exist numbers $r_0,s_0,\varepsilon_0>0$ such that for given s with $|s|\le s_0$ exactly one solution $\theta\varepsilon C_\nu(U)$ of (14) exists which satisfies the condition

$$||\theta||_\nu\le r_0,\ \ b_n(\theta)=s,\ \ |p-p_{n,b,q}|\le\varepsilon_0$$

For $s\neq 0$, p is uniquely determined: $p=p_{n,b,q}-\frac{n^2+b}{\pi_n(q)}$ while for s=0 only the trivial solution $\theta=0$ exists.

The solution can be given as an absolutely convergent series in $C_\nu(U)$

$$\theta=s\sin n\sigma+\sum_{\ell=2}^{\infty}s^\ell\,\theta_\ell \tag{17}$$

As $\tau_\ell\ =-X_q\theta_\ell$ the series

$$\tau\ =-\ \frac{s_n}{d_n}s\cos n\sigma+\sum_{\ell=2}^{\infty}s^\ell\,\tau_\ell \tag{18}$$

converges absolutely in $C_\nu(C)$. Also

$$p=p_{n,b,q}+\sum_{\ell=2}^{\infty}s^\ell\,\varepsilon_\ell \tag{19}$$

is absolutely convergent with $\varepsilon_{2m+1}=C$ (for the proof see [5]).

4. Construction of the Solutions

Using the method of Zeidler [5] we apply the operators T and J to the functional $a'(\theta,\tau)$ to get

$$a'=-1+\alpha\equiv -1+\sum_{\ell=1}^{\infty}s^\ell\,\alpha_\ell \tag{20}$$

Our problem can then be formulated as:

$$\theta=pJX_q\theta+bJ^2\theta+JN^*(\theta,\tau);\ a_0(N^*)=0$$

(21) with

$$N^*(\theta,\tau)=p(\tau-\sinh\tau)-\frac{p\alpha}{2}e^{-\tau}+b(e^{-\tau}Je^{-\tau}\sin\theta-J\theta)$$

Let $p=p_{n,b,q}+\varepsilon$. The pseudoresolvent $\tilde{R}_{n,b,q}$ will then be given as

$$\tilde{R}_{n,b,q}=(I-p_{n,b,q}\,JX_q-bJ^2-P_n)^{-1}$$

(22) with

$$P_n\theta=b_n(\theta)\sin n\theta$$

It follows that

$$\theta=\tilde{R}_{n,b,q}\,Jf$$

(23) with

$$f=-\varepsilon\tau\ +N^*(\theta,\tau)$$

We set (20) in (23) to get

$$f=\sum_{\ell=1}^{\infty}s^\ell\,f_\ell \tag{24}$$

and

$$\theta_\ell=\tilde{R}_{n,b,q}\,Jf_\ell,\quad \ell=2,3,\ldots \tag{25}$$

5. Computation of the Solutions

We consider the case n=1 of primitive waves, from which we can calculate the non-primitive waves n>1.

From (22) we have

$$k>1:b_k(\tilde{R}_{1,b,q}\,Jf)=\frac{k(d_{k+1}+d_{k-1})a_k(f)}{d_{k+1}(k-1)(k-b)+d_{k-1}(k+1)(k+b)}$$

(26)

$$\text{and}\ \ b_1(\tilde{R}_{1,b,q}\,Jf)=-a_1(f)$$

The algorithm consists of two steps:

(i) As $a_0(N^*(\theta,\tau))=0$ and $a_0(\tau)=0$, it follows from (23) that

$$a_0(f_\ell)=0 \tag{27}$$

from which we compute α_ℓ.

(ii) As $b_1(\theta)=s$, we have

$$b_1(\theta_\ell)=0,\quad \ell=2,3,\ldots \tag{28}$$

from which we compute $\varepsilon_{\ell-1}$.

Expanding $f(\theta,\tau)$ to the third order gives

$$f=-\varepsilon\tau-\frac{p}{2}\alpha+\frac{p}{2}\alpha\tau-\frac{p}{4}\alpha\tau^2-bJ\theta\tau-b\tau J\theta+bJ(\frac{\tau^2}{2}\theta-\frac{\theta^3}{6})+$$

(29)

$$+b\tau J\tau\theta+b\frac{\tau^2}{2}J\theta-\frac{p}{6}\tau^3+\ldots$$

with $p=p_0+\varepsilon\equiv(d_1/s_1)(1+b)+\varepsilon$.

Now, we consider the first three orders of the solutions:

(a) First Order. From (29) we have $f_1=\frac{p_0}{2}\alpha_1$. From (27) we have $\alpha_1=0$ and we already know that the linearized problem has the solution

$$\theta_1=\sin\sigma,\quad \tau_1=-\gamma_1\cos\sigma\ \ \text{with}\ \gamma_1=s_1/d_1 \tag{30}$$

(b) Second Order. As $\varepsilon_1=0$ ($\varepsilon_{2m+1}=0$), we have

$$f_2=-\frac{p_0}{2}\alpha_2-bJ\theta_1\tau_1-b\tau_1J\theta_1=-\frac{p_0}{2}\alpha_2-\frac{b\gamma_1}{2}-\frac{3b\gamma_1}{4}\cos 2\sigma$$

From (27) we get $\alpha_2=-b\gamma_1/p_0$, consequently $f_2=-(3b\gamma_1/4)(\cos 2\sigma)$. Using (23) we get

$$\theta_2 = \delta_2 \sin 2\sigma, \qquad \tau_2 = -\gamma_2\delta_2 \cos 2\sigma$$

(31) with

$$\delta_2 = \frac{3(s_3 + 3s_1)b}{2(d_3(b-2) - 3d_1(b+2))}, \quad \gamma_2 = s_2/d_2$$

(c) Third Order. From (29) we have

$$f_3 = -\varepsilon_2\tau_1 - \frac{p_o}{2}\alpha_3 \underset{2}{+} \frac{p_o}{2}\alpha_2\tau_1 \underset{3}{-} bJ(\theta_1\tau_2 + \tau_1\theta_2) - b\tau_2J\theta_1 - b\tau_1J\theta_2 + bJ(\frac{\tau_1^2}{2}\theta_1 - \frac{\theta_1^3}{6}) + \tau_1J\tau_1\theta_1 + \frac{b}{2}\tau_1^2J\theta_1 - \frac{p_o}{6}\tau_1^3$$

After rewriting all products in terms of trigonometric polynomials, it is sufficient to consider the cos terms:

$$f_3 = \cos\sigma\ (\varepsilon_2\gamma_1 - \frac{3}{4}\gamma_1\delta_2 b + \frac{b}{8} + \frac{\gamma_1^2}{8})$$

Beside those terms there will be constants and terms in cos 3σ.

From (28) we get $a_1(f_3) = 0$ which yields

$$\text{(32)} \quad \varepsilon_2 = \frac{3}{4}\delta_2 b - \frac{b}{8\gamma_1} - \frac{\gamma_1}{8}$$

The solution of our problem will then be

$$\text{(33)} \quad \begin{aligned} \theta &= s \sin\sigma + \delta_2 s^2 \sin 2\sigma + O(s^3) \\ \tau &= -\gamma_1 s \cos\sigma - \gamma_2\delta_2 s^2 \cos 2\sigma + O(s^3) \\ p &= \frac{1+b}{\pi_1(q)} + (\frac{3}{4}\delta_2 b - \frac{b}{8\gamma_1} - \frac{\gamma_1}{8})s^2 + O(s^4) \end{aligned}$$

This result agrees with our result in the case of infinite channels when $q \to 0$ (see [4]).

Through back transformation we can get all physical parameters as well as the equation of the free surface $y_o(x)$.

Simple integration will yield

$$\text{(34)} \quad y_o(x) = \frac{\lambda}{2\pi}\{s \cos\frac{2\pi}{\lambda}x + \tfrac{1}{2}(\delta_2 + \frac{\gamma_1}{2})s^2 \cos\frac{4\pi}{\lambda}x + O(s^3)\}$$

REFERENCES

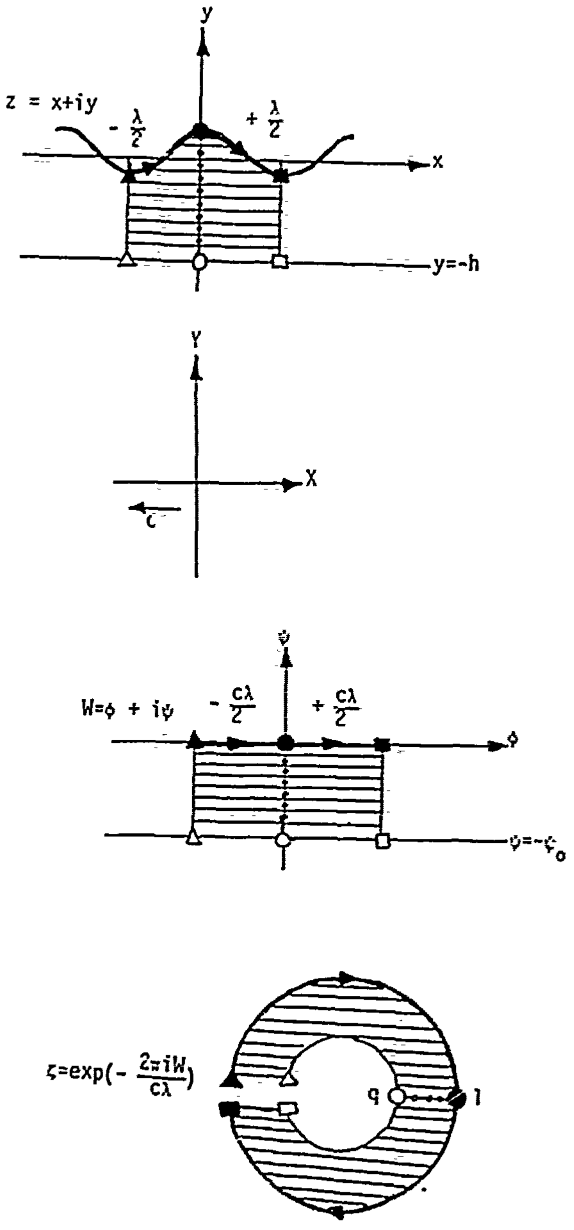
[1] Fatou, P.: Séries trigonometriques et séries de Taylor, Acta Math. 30, 355-400 (1906).

[2] Levi-Civita, T.: Détermination rigoureuse des ondes permanentes d'ampleur finie. Math. Ann. 93, 264-314 (1925).

[3] Moussa, M.: A general computational method for solving nonlinear free boundary value problems of plane hydrodynamics, to appear in Communications in Applied Numerical Methoαs.

[4] Moussa, N.: On permanent capillar-heavy waves in infinite channels, submitted for publication.

[5] Zeidler, E.: Beitraege zur Theorie und Praxis freier Randwertanfgaben. Akademic Verlag, Berlin (1971).



$$\zeta = \rho e^{i\sigma}$$

$$\rho = \exp(\frac{2\pi\psi}{c\lambda})$$

$$\sigma = -\frac{2\pi\phi}{c\lambda}$$

$$q = \exp(\frac{2\pi\psi_o}{c\lambda}) \le |\zeta| \le 1$$

Figure 1

COMPUTATION OF TRANSONIC FLOW BY PASSIVE BOUNDARY-LAYER CONTROL ON AN AEROFOIL.

by

R.K. Cooper, S. Raghunathan and J.L. Gray,
Department of Aeronautical Engineering,
The Queen's University of Belfast,
Belfast, Northern Ireland.

Abstract - A computational method used to calculate the transonic flow over an aerofoil with passive boundary-layer control is described briefly. Both the computational and experimental results show that passive control can reduce both viscous and wave drag of aerfoils in transonic flow.

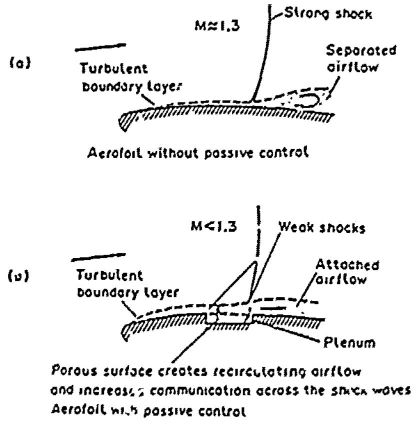
1. INTRODUCTION

Transonic flow over an aerofoil contains supersonic regions embedded in a subsonic field. The supersonic flow invariably terminates in a shock wave. The formation of shock waves on an aerofoil gives an additional pressure drag (wave drag) to the subsonic pressure drag. The shock wave also imposes an adverse pressure gradient on the boundary-layer.

A control technique which appears to show some promising results for a substantial drag reduction in transonic flow is the passive control of shock-boundary-layer interaction[1-2](PCSB). This paper presents a brief review of computational approaches to the solution of transonic flow with passive boundary-layer control.

2. THE CONCEPT OF PASSIVE CONTROL OF SHOCK-BOUNDARY LAYER INTERACTION

The concept of PCSB as originally suggested by Bushnell and Whitcomb (see Ref.1) consists of a porous surface and a cavity or plenum underneath located in the region of shock boundary-layer interaction (Fig. 1). It is suggested that the static pressure rise across the shock wave will result in a flow through the cavity from downstream to upstream of the shock wave. This is equivalent to a combination of suction downstream and blowing upstream of the shock. The cavity would also increase the communication of signals across the shock wave. These effects would lead to a rapid thickening of the boundary-layer approaching the shock which in turn should produce a system of weaker shocks with an extended interaction region. This will reduce the wave drag. The suction downstream of the shock can also reduce separation and therefore reduce viscous losses. The porous surface and the cavity can also damp pressure fluctuations associated with shock wave boundary-layer interaction.



Aerofoil without passive control

Aerofoil with passive control

Fig. 1.The concept of passive boundary-layer control

The conservation form of the two dimensional compressible Navier-Stokes equation in cartesian co-ordinates without body forces or external heat transfer can be written as

$$\frac{\partial u}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \qquad (1)$$

where the vector equation comprises conservation of mass, momentum and energy.

The approximations made to the above equations to reduce computational grids and time in the order of reducing accuracy are (i) thin layer approximations (TLA), (ii) parabolised Navier-Stokes (PNS), (iii) Euler equations, (iv) full potential flow equations and (v) small perturbation equations. Whereas (i) and (ii) resolve viscous terms in the flow to some degree of approximation, (iii) to (v) neglect the viscous terms. The viscous terms can be resolved by coupling (iii), (iv) or (v) with equations based on boundary-layer approximations.

3. COMPUTATION OF TRANSONIC FLOW WITH PASSIVE CONTROL

Some of the methods mentioned above have been used to compute transonic flow over an aerofoil with passive control. These are described briefly here.

3.1 Transonic small disturbance approach

The equation describing two dimensional invicid irrotational transonic flow around a thin aerofoil in transformed ξ, η co-ordinates in non conservative form can be expressed as

$$\{K\phi^*_\xi - \frac{\gamma+1}{2} M^2_\infty \phi^{*^2}_\xi\}_\xi - \phi_{\eta\eta} = 0 \qquad (2)$$

where $K = (1-M^2_\infty)/\tau^{2/3}$

$$\phi^* = \phi/(cv_\infty),\ \tau = t/c,\ \xi = x/c$$

$$\eta = y/\{c\tau^{-1/3}\}$$

where ϕ is velocity potential, c is aerofoil chord, τ is thickness ratio, M_∞ free stream Mach number.

The boundary condition on a solid surface is

$$\phi^*_\eta = 0 \qquad (3)$$

The equation can be solved by a successive line over-relaxation procedure with a central difference scheme in the subsonic region (elliptic) and upwind difference scheme in the supersonic region (hyperbolic)[3]. Savu and Trifu[4] used this procedure to calculate the flow field around a porous aerofoil assuming that the normal velocities in the porous region obey Darcy's law

$$v_\eta = \sigma \Delta p, \quad \sigma = \bar{\sigma}/(\rho_\infty V_\infty)$$

where $\bar{\sigma}$ is the porosity factor.

3.2 Full potential flow approach

Subsequently Chen et al[5] used full potential flow equations with the same boundary conditions used by Savu and Trifu to solve this problem. The results of the computation agreed qualitatively with those of Savu and Trifu. Further, it was shown that passive control can not only reduce drag but augment lift on an aerofoil.

3.3 Interactive boundary-layer approach

Refinement to the above solutions was achieved by introducing viscous effects. These include for the flow around an aerofoil with passive control, thin layer approximations by Chung et al[6], coupling full potential outer flow to a boundary-layer equation near the surface by Chung et al[6] and coupling transonic small perturbation theory to boundary-layer equations near the surface by Gray[7].

The viscous flow near the surface of the aerofoil and wake can be expressed by the following non dimensional non conservative steady two dimensional boundary-layer equations in transformed ξ, η co-ordinates.

$$(\rho u)_\xi \, \xi_x + (\rho v)_\eta \, \eta_x = 0 \tag{4}$$

$$\rho\{u(u_\xi \, \xi_x + u_\eta \, \eta_x) + v \, u_\eta \, \eta_y\} =$$

$$- \beta \, p_\xi \, \xi_x + (\mu \, u_\eta \, \eta_y)_\eta \, \eta_y \tag{5}$$

$$\rho c_p\{u(T_\xi \, \xi_x + T_\eta \, \eta_x) + v \, T_\eta \, \eta_y\} =$$

$$\beta u \, p_\xi \, \xi_x + (k \, T_\eta \, \eta_y)_\eta + \mu \, (u_\eta \, \eta_y)^2 \tag{6}$$

$$p = \rho T, \quad \beta = p/\rho u^2, \quad c_p = \gamma R/(\gamma - 1)$$

The boundary-layer equations were solved using MacCormack's predictor-corrector algorithm with appropriate initial conditions.

Baldwin-Lomax turbulence model was used for the transonic small perturbation boundary-layer interaction code.

Typical results obtained from these computations are shown in Fig. 2. The computational results agree qualitatively with the experimental results[2] and show some of the features of transonic shock wave boundary-layer interaction with passive control. The porous surface splits the flow into a λ shock with the leading edge of the shock system anchored to the beginning of the porous region. This has the effect of reducing entropy changes across the shock and therefore the wave drag. It has also been shown that passive control can alleviate shock oscillations in the transonic flow.

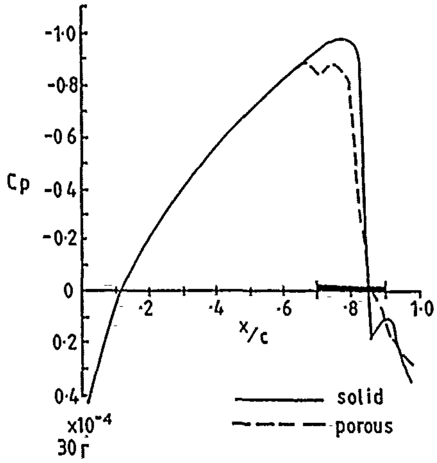


Fig. 2. Pressure distribution $\tau = 12\%$ circular arc $M_\infty = 0.83$, $R = 2 \times 10^6$, $\sigma = 0.03$

References

1. Bahi, L., Ross, J.M. and Nagamatsu, H.T., Passive shock wave/boundary-layer control for transonic aerofoil drag reduction. AIAA paper 83-0137, 1983.

2. Raghuanthan, S., Passive control of shock boundary-layer interaction. Prog. Aerospace Sci. Vol. 25, 1988. pp 271-296.

3. Murman, E.M. and Cole, J.D., Calculation of plane steady transonic flow. AIAA Jl., Vol. 9, Jan 1971. pp 114-121.

4. Savu, G. and Trifu, O. Porous aerofoils in transonic flow. AIAA Jl. Vol. 22, No. 7, 1984 pp 989-991.

5. Chen C.L., Chow, S.Y., Holst, T.L. and Van Dalsem, W.R., Numerical simulation of transonic flow over porous aerofoils. AIAA paper, 85-5022, 1985.

6. Chung, L.C., Chuen, Y.C., Van Dalsem, W.R. and Holst, T.L. Computation of viscous transonic flow over porous aerofoils. AIAA paper 87-0359, 1987.

7. Gray, J.L. The passive control of shock wave boundary-layer interaction in transonic flow. PhD thesis, Queen's University of Belfast, 1989.

PRECONDITIONING OF DISCRETIZED PARABOLIC PROBLEMS ON TWO-LEVEL GRIDS WITH LOCAL REFINEMENT

Richard E. Ewing
Department of Mathematics, Chemical Engineering,
and Petroleum Engineering
University of Wyoming
Laramie, Wyoming 82071

Panyot Vassilevski
Department of Mathematics
University of Wyoming
Laramie, Wyoming 82071

Abstract–We describe an efficient preconditioning technique for solving time dependent problems with local refinement both in time and in space. The preconditioner makes use of solvers on regular grids only, hence it can be easily incorporated in existing codes for solving parabolic problems based on standard timestepping. The difficulty that arises in local timestepping is that such a discretization always leads to nonsymmetric matrices. However, it turns out that we can precondition these problems with generalized conjugate gradient (GCG) type methods in an optimal way.

I. INTRODUCTION

We describe an efficient solution technique based on Domain Decomposition (DD) ideas for solving time dependent diffusion problems. We consider the following model problem

$$\frac{\partial p}{\partial t} - \nabla \cdot \left(\frac{k}{\mu}\nabla p\right) = q \text{ in } \Omega \subset \mathbb{R}^2, \quad t \geq 0,$$

with appropriate boundary and initial conditions. The discretization of the problem can be done in the framework of the discontinuous Galerkin method, (cf. Thomeé [6]) where we have a global time step τ_c. In the time interval $(t_n, t_{n+1}]$, $t_n = n\tau_c$, $n \geq 0$, in regions $\Omega_2 \subset \Omega$ of special interest, we introduce a finer time step $\tau_f = \tau_c/m$ for some $m > 1$. We also couple this with local refinement in space on the subdomain Ω_2. See Figure 1 for a one dimensional domain Ω. Then we have intermediate time levels $t_{n,i} = t_n + i\tau_f$, $i = 0, 1, \ldots, m$.

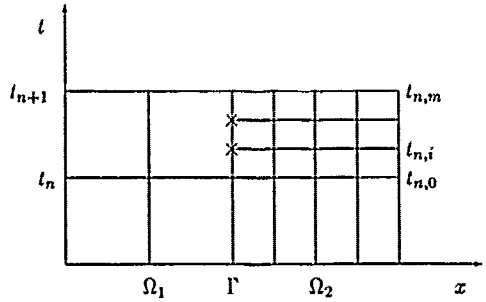


Figure 1. Grid with local refinement in space and in time. At the interface $\Gamma \times (t_n, t_{n+1}]$ 'slave' nodes (denoted by '×') are introduced.

By using a variational formulation (for details see Ewing *et al.* [5]) or by cell-centered finite difference approximation of the above problem, cf. Ewing *et al.* [3], one can derive a composite grid problem for the unknowns between two global time steps t_n and t_{n+1}. We obtain a linear algebraic problem of the form, $Ax = b$. Note that, the composite-grid matrix obtained in this way is always nonsymmetric, but has a coercive symmetric part.

II. TWO-GRID PRECONDITIONER

Using the following DD ordering of the nodes in the time slab $(t_n, t_{n+1}] \times \Omega$,

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}_f \\ \mathrm{x}_c \end{bmatrix} \begin{matrix} \}\Omega_2 \times (t_n, t_{n+1}] \\ \}\Omega_1 \times \{t_{n+1}\} \end{matrix},$$

we obtain the following 2×2 block form of A,

$$A = \begin{bmatrix} A_f & B \\ C & D \end{bmatrix},$$

where the block A_f corresponds to a standard timestepping procedure on the subdomain $\Omega_2 \times (t_n, t_{n+1}]$. That is, to solve systems defined by A_f, we can use any available timestepping code on a rectangular grid.

The preconditioner, originally proposed in Bramble *et al.* [2] for elliptic problems with local refinement, and extended for time dependent problems in Ewing *et al.* [3–5], is constructed for the reduced problem

$$S\mathrm{x}_c = \mathrm{b}_c - CA_fB_f,$$

where $S = D - CA_f^{-1}B$ is the reduced Schur matrix. Consider the problem on the original coarse-grid (i.e., without any local refinement in either space or time). Then we have a coarse-grid matrix $\tilde{A}$ defined for the coarse grid at the time level $t = t_{n+1}$. Formally, using the same DD idea, we can partition $\tilde{A}$ in 2×2 block form as follows

$$\tilde{A} = \begin{bmatrix} \tilde{A}_2 & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} \begin{matrix} \}\Omega_2 \times \{t_{n+1}\} \\ \}\Omega_1 \times \{t_{n+1}\}, \end{matrix}$$

where $\tilde{A}_2$ is a block-matrix corresponding to a global timestepping but for the subdomain Ω_2 only. Note that in this case $\tilde{A}$ is symmetric and positive definite, hence its reduced Schur matrix $\tilde{S} = \tilde{D} - \tilde{C}\tilde{A}_2\tilde{B}$ is also symmetric and positive definite. The preconditioner for the reduced problem we choose is $\tilde{S}$. Note that in order to solve a system with $\tilde{S}$, we can use solvers for the matrix $\tilde{A}$, hence we can use any available software for timestepping procedures on rectangular grids. In Ewing *et al.* [4] it was shown that using an approximate $\tilde{S}$ in a GCG-type method (since S is nonsymmetric) will give an optimal convergent method. The convergence properties are independent of possible jumps of the coefficient k/μ and the coarse-grid sizes. For more details and some numerical experiments we refer to Ewing *et al.* [4,5]. For application in industrial reservoir simulation codes, see Boyett *et al.* [1].

REFERENCES

1. B.A. Boyett, M.S. El-Mandouh, and R.E. Ewing, Local grid refinement for reservoir simulation, *Mathematical and Computational Issues in Geophysical Fluid and Solid Mechanics*, SIAM, Philadelphia, PA, (to appear).

2. J.H. Bramble, R.E. Ewing, J.E. Pasciak, and A.H. Shatz, A preconditioning technique for the efficient solution of problems with local grid refinement, *Comp. Meth. Appl. Mech. Eng.* 67 (1988), 149–159.

3. R.E. Ewing, R.D. Lazarov, and P.S. Vassilevski, Finite difference schemes on grids with local refinement in time and space for parabolic problems I. Derivation, stability, and error analysis, *Computing* 45 (1990), 193–215.

4. R.E. Ewing, R.D. Lazarov, and P.S. Vassilevski, Finite difference schemes on grids with local refinement in time and in space for parabolic problems II. Optimal order two-grid iterative methods, *Proceedings of the VIth GAMM Seminar on Parallel Methods for PDEs* (W. Hackbusch, ed.), January 19–21, 1990, Vieweg, Wiesbaden, 70–93.

5. R.E. Ewing, R.D. Lazarov, J.E. Pasciak, and P.S. Vassilevski, Finite element method for parabolic problems with time steps variable in space, (in preparation).

6. V. Thomeé, Galerkin finite element methods for parabolic problems, *Lect. Notes on Math.*, Springer, 1984.

A DOMAIN-DECOMPOSITION PROCEDURE FOR PARABOLIC EQUATIONS

CLINT DAWSON
Department of Mathematical Sciences
Rice University
Houston, TX 77251-1892 U.S.A.

Abstract A domain decomposition procedure for solving parabolic partial differential equations based on explicit/implicit, Galerkin finite element discretizations is presented. Error estimates are stated, and numerical results on a parallel processing machine are discussed.

I. INTRODUCTION

Domain decomposition procedures have received much attention in recent years as a way of numerically solving partial differential equations on parallel processing machines, see, for example, [1]. Much of this work has been directed at elliptic equations. These techniques are often extendable to implicit time discretizations of parabolic equations, since an "elliptic" equation must be solved at each time step, generally resulting in an algorithm requiring iteration between subdomain problems and interface problems. In the work described here, we avoid this complication by using information from the previous time step to calculate fluxes along an interface between subdomains. These approximate fluxes are then used as boundary data for implicit subdomain problems. Thus, the procedure is noniterative and uses nonoverlapping subdomains. For the method described here, a Galerkin finite element discretization is used in each subdomain

II. ALGORITHM DESCRIPTION

Let Ω denote a spatial domain in $\mathbf{R}^d$. Denote by $H^m(\Omega)$ and $W^m_\infty(\Omega)$ the standard Sobolev spaces on Ω, with norms $\|\cdot\|_m$ and $\|\cdot\|_{\infty,m}$, respectively. Let $L^p(\Omega)$, $p = 2, \infty$, denote the standard Banach spaces, with $\|\cdot\|$ denoting the L^2 norm, $\|\cdot\|_\infty$ the L^∞ norm.

Let $[\alpha,\beta] \subset [0,T]$ denote a time interval, $X = X(\Omega)$ a normed space. To incorporate time dependence, we use the notation $\|\cdot\|_{L^p(\alpha,\beta;X)}$ to denote the norm of X-valued functions f with the map $t \mapsto \|f(\cdot,t)\|_X$ belonging to $L^p(\alpha,\beta)$.

First, consider the case $d = 2$, and $\Omega = (0,1)\times(0,1)$. Decompose Ω into two subdomains, $\Omega_1 = (0,\frac{1}{2})\times(0,1)$, and $\Omega_2 = (\frac{1}{2},1)\times(0,1)$. Let Γ denote the interface between these two domains, $\Gamma = \{\frac{1}{2}\}\times(0,1)$, and let u satisfy

$$u_t - \Delta u + u = 0, \quad \text{on } \Omega\times(0,T], \tag{1}$$

$$u(x,0) = u^0(x), \quad \text{on } \Omega, \tag{2}$$

$$\frac{\partial u}{\partial n_\Omega} = 0, \quad \text{on } \partial\Omega\times(0,T], \tag{3}$$

where n_Ω is the outward normal to $\partial\Omega$.

Our domain decomposition procedure relies on calculating an approximate normal derivative of u on Γ. Let $\phi(x) = \phi_2((x-1/2)/H)/H$, where H is a parameter, $\frac{1}{2} \geq H > 0$, and

$$\phi_2(x) = \begin{cases} 1-x, & 0 \leq x \leq 1, \\ x+1, & -1 \leq x \leq 0, \\ 0. & \text{otherwise} \end{cases} \tag{4}$$

For ψ a smooth function, define an approximate normal derivative of ψ at each point y on Γ by

$$\psi_x(\frac{1}{2},y) \approx B(\psi)(\frac{1}{2},y) = -\int_0^1 \phi'(x)\psi(x,y)dx, \tag{5}$$

and note that

$$|\psi_x(\frac{1}{2},y) - B(\psi)(\frac{1}{2},y)| \leq CH^2\|\psi_{xxx}\|_\infty. \tag{6}$$

Let $0 = t^0 < t^1 < \ldots < t^M = T$ be a given sequence; $\Delta t^n = t^n - t^{n-1}$. We approximate u at time t^n by U^n, where $U^n|_{\Omega_j} \in \mathcal{M}_j$, and $\mathcal{M}_j$ is a finite dimensional subspace of $H^1(\Omega_j)$. Let $\mathcal{M}$ be the subspace of $L^2(\Omega)$ such that if $v \in \mathcal{M}$, $v|_{\Omega_j} \in \mathcal{M}_j$, and let $[v]$ denote the jump in function values of v across Γ (which is well-defined). The approximations $U^1,\ldots,U^M$ are given by

$$\sum_{j=1}^{2}\left\{(\partial_t U^n, v)_{\Omega_j} + D_j(U^n,v)\right\} + (B(U^{n-1}),[v])_\Gamma = 0, \quad v \in \mathcal{M}. \tag{7}$$

Here $(\cdot,\cdot)_\Lambda$ denotes the $L^2(\Lambda)$ inner product,

$$D_j(U^n,v) = (\nabla U^n, \nabla v)_{\Omega_j} + (U^n,v)_{\Omega_j},$$

and $\partial_t U^n = (U^n - U^{n-1})/\Delta t^n$. Note that once $B(U^{n-1})$ has been calculated, U^n can be computed on Ω_1 and Ω_2 completely independently

In order to state an error estimate for the above algorithm, let $W \in \mathcal{M}$ be the elliptic projection of u, defined for each $t \in [0,T]$ by

$$\sum_{j=1}^{2} D_j((W-u)(\cdot,t),v) = 0, \quad v \in \mathcal{M}. \tag{8}$$

Let $\eta = u - W$. The following theorem is proved in [2].

Theorem 1 *Suppose that the solution u is sufficiently smooth and that $U^0 \in \mathcal{M}$ is taken to be $W(\cdot,0)$. Let $\Delta t = \max_n \Delta t^n$. Then there exists a constant C, independent of the spaces $\mathcal{M}_j$, such that*

$$\max_n \|u^n - U^n\| \leq C\left(\Delta t + H^{2.5} + \int_0^T \|\eta_t(\cdot,t)\|dt + H^{-\frac{1}{2}}\|\eta\|_{L^\infty(\Omega\times(0,T))}\right),$$

provided that $\Delta t \leq \frac{H^2}{4}$.

The algorithm can be extended to domains $\Omega \in \mathbf{R}^d$ with piecewise uniformly smooth, Lipschitz boundaries, where the interface Γ is a uniformly smooth, $(d-1)$-dimensional manifold. Moreover, different functions other than ϕ_2 may be used in the flux approximation B, which give higher order accuracy in H. In particular, suppose $\phi(x) = \phi_4((x-1/2)/H)/H$, where

$$\phi_4(x) = \begin{cases} (x-2)/12, & 1 \leq x \leq 2, \\ -5x/4 + 7/6, & 0 \leq x \leq 1, \\ 5x/4 + 7/6, & -1 \leq x \leq 0, \\ -(x+2)/12, & -2 \leq x \leq -1, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

In this case, for ψ five times differentiable in x,

$$|\psi_x(\frac{1}{2},y) - B(\psi)(\frac{1}{2},y)| \leq CH^4. \tag{10}$$

The proof of Theorem 1 relies on the following coercivity property of the method. Namely, for $\psi \in \mathcal{M}$, let

$$|||\psi|||^2 \equiv \sum_{j=1}^{2} D_j(\psi,\psi) + \frac{1}{H}\|[\psi]\|^2_\Gamma, \tag{11}$$

where

$$\|[\psi]\|^2_\Gamma = ([\psi],[\psi])^2_\Gamma.$$

Then, under the assumptions of Theorem 1, it can be shown that

$$|||\psi|||^2 \leq C\left(\sum_j D_j(\psi,\psi) + (B(\psi),[\psi])_\Gamma\right). \tag{12}$$

In general, assume that for some $H > 0$, B is a linear map of $L^2(\Omega)$ into $L^2(\Gamma)$ and that it satisfies the following four conditions.

(i) There is a constant C_0 such that

$$|||\psi|||^2 \leq C_0(D(\psi,\psi) + (B(\psi),[\psi])_\Gamma), \tag{13}$$

(ii) There is a constant C_1 such that

$$\|B(\psi)\|^2_\Gamma \leq C_1 H^{-3}\|\psi\|^2. \tag{14}$$

(iii) There is a constant C_2 such that

$$\|B(\psi)\|^2_\Gamma \leq C_2 H^{-1}\|\psi\|_\infty. \tag{15}$$

(iv) There is a constant $k \geq 0$ and a constant C_3 which depends on the solution u such that

$$\left\|\frac{\partial u(\cdot,t)}{\partial\gamma} - B(u)(\cdot,t)\right\|_\Gamma \leq C_3 H^k, \tag{16}$$

for $0 \leq t \leq T$, where $\partial u/\partial\gamma$ is the normal derivative of u on Γ, in the direction from Ω_1 to Ω_2.

We note that for $\phi = \phi_2$, Ω the unit square on $\mathbf{R}^2$, and $\Gamma = \{\frac{1}{2}\} \times (0,1)$, $C_0 = 1.7$, $C_1 = 2$, and $C_3 = 2$. For $\phi = \phi_4$, $C_0 = 1.64$, $C_1 = 3.14$, and $C_2 = 8/3$.

In general, assuming conditions (i)-(iv) hold, we have the following Theorem [2].

Theorem 2 *Suppose that u is sufficiently smooth and that $U^0 = W(\cdot, 0)$. Let $\Delta t = \max_n \Delta t^n$. Then there exists a constant C, independent of the spaces $\mathcal{M}_j$ such that*

$$\max_n \|(u^n - U^n)\| \leq C \left(\Delta t + H^{k+\frac{1}{2}} + \int_0^T \|\eta_t(\cdot, t)\| dt + H^{-\frac{1}{2}} \|\eta\|_\infty \right),$$

provided

$$\Delta t \leq \frac{H^2}{C_0 C_1}. \tag{17}$$

III. NUMERICAL EXAMPLES

We have implemented the algorithm outlined above on a 32 processor Intel iPSC/860 at the Center for Research in Parallel Computing, Rice University. We now present some timings which demonstrate the parallelism of the method.

The first problem tested was

$$u_t - \Delta u = 0, \quad \text{on } \Omega \times (0, T], \tag{18}$$

$$u(x, 0) = \cos(\pi x)\cos(\pi y), \quad \text{on } \Omega, \tag{19}$$

$$\frac{\partial u}{\partial n_\Omega} = 0, \quad \text{on } \partial\Omega \times (0, T], \tag{20}$$

with Ω the unit square on $\mathbf{R}^2$. This equation has solution $u(x, y, t) = e^{-2\pi^2 t} u^0(x, y)$.

In our domain decomposition procedure, once the interface fluxes are calculated, a system of linear algebraic equations must be solved within each subdomain. Preconditioned conjugate gradient with diagonal preconditioning was used to solve these subdomain problems.

In Table 1, we present timings for runs with different types of domain decompositions. In column 1 of Table 1, the notation "$m \times n$" refers to a decomposition of Ω into m subdomains in the x direction, and n in the y direction. The decomposition was done so that the subdomains contained the same number of unknowns. In these runs, $\Delta t = .0025$ and $H = .10$. The final time $T = .10$. The approximating space in each subdomain was the tensor product of continuous piecewise linear functions in x and y. An underlying rectangular, uniform, 80-by-80 grid was used. We present both the CPU time for each case and the average number of conjugate gradient iterations. The latter number was computed by summing the number of conjugate gradient iterations (per subdomain) required to "solve" the linear algebraic system at each time step, and dividing this quantity by the number of time steps. ("Solving" the linear algebraic system meant reducing the residual below 10^{-6}.) In general, this number is subdomain dependent, however, for this particular problem, it was essentially the same on each subdomain. Not surprisingly, however, the number varied with the decomposition. The timings presented in Table 1 indicate that, for a fixed number of unknowns, the run time essentially decreased by a factor of two when the number of processors was doubled.

The second problem we considered was

$$u_t - \Delta u = 0, \quad \text{on } \Omega \times (0, T], \tag{21}$$

$$u(x, 0) = 0, \quad \text{on } \Omega, \tag{22}$$

Decomposition	CPU (sec)	C. G. iter.
2x1	106.650	41
2x2	55.010	48
4x2	27.819	47
4x4	10.642	37

Table 1: TIMINGS ON AN Intel iPSC/860: PROBLEM 1

Decomposition	CPU (sec)	C. G. iter.
2x1	110.608	43
2x2	49.391	42
4x2	25.435	42
4x4	9.696	33

Table 2. TIMINGS ON AN Intel iPSC/860. PROBLEM 2

with

$$\frac{\partial u}{\partial n_\Omega} = \begin{cases} 1, & \text{on } \{0\} \times (0,1) \times (0,T], \\ -1, & \text{on } (1/4, 1) \times \{0\} \times (0,T], \\ 0, & \text{otherwise}. \end{cases} \tag{23}$$

This problem corresponds to having a heat source on the left boundary of Ω and a heat sink on the bottom boundary. The results in this case are given in Table 2. For this problem, the average number of conjugate gradient iterations varied from one subdomain to the other. In Table 2, we present the maximum of these averages. This is the quantity which controls the problem, since data cannot be passed from one subdomain to another until the linear algebraic problems in each subdomain have converged. As in the previous case, the computation time is reduced by a factor of two with each doubling of processors. A factor of greater than two is obtained when the number of conjugate gradient iterations decreases.

REFERENCES

[1] T. F. Chan, R. Glowinski, J. Periaux, and O. Widlund (editors), *Domain decomposition algorithms*, Proceedings of the 2nd International Conference on Domain Decomposition Methods, Jan. 14-16, 1988, UCLA, SIAM Publications, 1989.

[2] C. N. Dawson and T. F. Dupont, *Explicit/implicit, conservative, Galerkin domain decomposition procedures for parabolic problems*, to appear in Math. Comp.

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures*

Serge G. Petiton

Institut de Calculs Mathématiques
Université Paris 7
75251 Paris Cedex, FRANCE
and
Department of Computer Science
Yale University
P.O. Box 2158, Yale Station
New Haven, CT 06520, USA

Christine J. Weill-Duflos

Laboratoire MASI
Université P. et M. Curie (Paris 6)
75252 Paris Cedex, FRANCE.

ABSTRACT

Computing Sparse Linear Algebra problems on SIMD massively parallel architectures is an important and major challenge for the future of these machines. In order to obtain good performance when solving many large sparse linear algebra problems, we use a communication compiler which permits to obtain a good speed-up compared to classical global communications. We study in this paper a polynomial preconditioned conjugate gradient method to solve a symmetric positive definite very sparse linear system on massively parallel architectures. We evaluate performance obtained on CM-2 with respect to a few very sparse matrices. We conclude that to solve some large very sparse matrix linear algebra problems we can now rely on CM-2 performance as fast as could be obtain on vector machines for dense problems a few years ago.

1 Introduction

Researches in massively parallel algorithms are numerous. Especially since the introduction of the Connection Machine 2 (CM-2) by Thinking Machines Corporation. Sparse computations are not yet very developed despite the supercomputer target applications developed on such problems.

The sizes of the concerned sparse matrices are sometimes very large and the degree of row (resp. column), ie the number of no zero elements by row (resp. column), is often small. Many scientific research fields generate now very large linear algebra problems with row degrees smaller than one per thousand of the matrix order. Thus, in scientific computing we often need to solve a linear system starting with very large sparse matrices. We shall study in this paper unstructured very sparse matrices, i.e. with very small degree of row and column.

Iterative methods are often used to solve these problems on SIMD massively parallel machines. These methods often use 3 major operations. sequential scalar operations, reductions (principally inner products) and matrix vector operations (principally matrix vector multiplication). Scalar operations are often made redundant on each processor to optimize communications betw.zen processors.

2 Sparse Matrix Computation on the CM-2

On many iterative linear algebra methods when the matrices are sparse, the matrix-vector multiplication is the only computation that manipulates sparse objects [4]. Thus, we first focus on the massively parallel sparse matrix-vector multiplications.

Methods to compute the multiplication of a sparse matrix by a vector on massively parallel architectures principally depend on the matrix and vector mapping and on the number of processors Let us assume that n is the matrix order and C the degree of row that we suppose equal to the degree of column Thus, with Cn processors we can map the sparse matrices with scan class as described by P. Kumar [3], with only one element of the matrix by virtual processor, see [7] for detailed implementation on the CM-2.

The idea is to compute all the floating point multiplications of the matrix-vector operation in massively parallel mode and to reduce with addition the other operations in a $\log_2(C)$ theoretical complexity. But we need two different mappings to do that efficiently. Then, we need to change the mapping during the matrix-vector operation. The multiplication will be done with a mapping in wh'ch the element $a(i,j)$ will be mapped on the virtual processor (ic,j), $ic = 0, C-1$ and $j = 0, n-1$, where ic is the row indice on the Compress Sparse Column (CSC) format representation of the matrix, see [5] for the description of the sparse matrix formats. Then, if we want to reduce with addition, along only one dimension, the partial results in a log arithmic complexity, we need to re-map the partial results, stored on virtual processors (ic,j) to the virtual processor (jc,i), where jc is the column indice of the element $a(i,j)$ on the Compress Sparse Row (CSR) format representation. Thus, the complexity of this column oriented sparse matrix vector multiplication is $1 + log_2(C)$ operations plus one global one-to-one general send operation

With the general CM-2 router, the time to do these irregular communications is really a strong bottleneck, except for very well adapted sparse matrix patterns. Fast sparse computation is really impossible on the CM-2 without using special tools to optimize these one-to-one general communications. Denny Dahl developed such a tool [2] which permits us to obtain more than a speed up of 10 compared to the CM-2 general router [7]. Then, we can really do sparse computations on the CM-2.

It is an important and difficult goal to propose test sparse ma-

*This research was supported in part by the Direction Recherches et Etudes Techniques, French Department of Defense, under contract 20.357/91/ETCA/CREA while the authors were in residence at the Etablissement Technique Central de l'Armement (ETCA).

trices to evaluate the performance for massively parallel algorithm computations. In this paper we do not propose special application patterns. We try to find bounds on the performance. These ones depend principally on the global send and especially on the distance between the diagonal of the sparse matrices and the C non-zero elements. Then, we take matrices with C non-zero diagonals around the main diagonal (C-diagonal matrix pattern) and we perturb at random them with a given probability q, which is a parameter of the evaluation. Thus, it is nothing but the exchange of non-zero elements (i,j) of the C-diagonal with the elements $(i,jrand)$, where $jrand$ is a random integer number between 0 and $n-1$. Each of these perturbations of the C band matrix is done with the probability q. Hence, we can study the performance with respect to q. We also use matrices with non-zero elements uniformly distributed all along the rows and columns, with a non-zero main diagonal (C distributed matrix pattern). In the first case we will obtain close-to-peak performance when $q = 0$, i.e. without perturbation of the C diagonals, for this algorithm and in the other case we will have the lower performance.

But, this tool uses the CM-2 as a hypercube of the vector accelerators. As we have just one vector accelerator per each 32 physical processors set, for example on a 16K CM2 we have 512 of these floating point fast units, can we conclude that it is always *massively* parallel computation?

3 Algorithm and Implementation

We are interested in the conjugate gradient method for the resolution of the linear system $Ax = b$ where A is a very sparse symmetric positive matrix. To improve the conjugate gradient method, it is possible to combine it with a polynomial preconditioning, see [1] and [6].

The polynomial s we chose for the preconditioning, is given by the Neuman serie :
$s(A) = I + B + \cdots + B^m$ where $B = I - A$ and $\|B\| < 1$.

The through-put, in term of Megaflops (Mflops), does not depend on the coefficients of the polynomial but on the degree.

We map the vector and scalar on a 2 D C x n grid compatible with the mapping of the matrix and to optimize communications. We often compute with redundancy : for example the inner product is computed on each row of processors and distributed on each processor. And, we do not compute the polynomial directly but we use the matrix-vector multiplication to compute $s(A)x$ at each iteration of the algorithm.

Likewise the matrix-vector multiplication, we take the same test matrices to obtain the performance of the algorithm. Figure 1 shows the performance on a 16K CM-2 when the degree of the polynomial is 4 and the matrix coefficients are real. We observe that the performance decreases as q increases and that the uniformly distributed pattern matrix is really a lower-bound. The performance looks like the matrix-vector multiplication performance. The performance we take, proves that the computational rate increases with the degree of the polynomial.

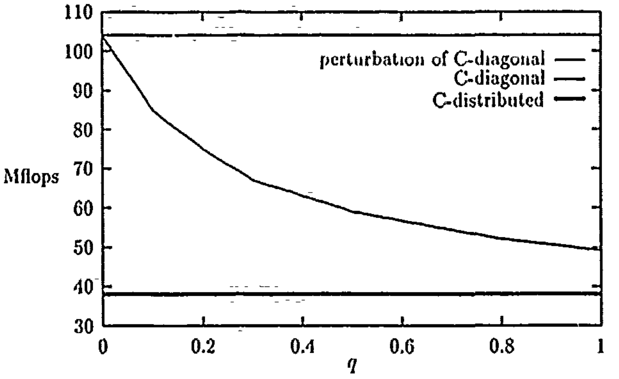


Figure 1: Polynomial Preconditioned Conjugate Gradient performance on the CM-2. $C = 4$ and $n = 128K$.

4 Conclusion

We show that sparse computations are possible on massively parallel architectures. We assume in our method that $Cn \geq p$ to have a $vpr \geq 1$ but we do not need $n \geq p$ (where p is the number of physical processors). Problems from fluid mechanics, quantum chemistry or biology generate such problems. For dense or sparse computations, parallel machines will become more and more powerful. Teraflops peak performance and giga-word memories will soon allow us to access a new area of scientific computing. The challenge will be to propose efficient parallel algorithms for these machines. The researches presented in this paper are a step in this direction.

References

[1] S. Ashby, T. Manteuffel and P. Sayor, *A Taxonomy for Conjugate Gradient Methods*, SIAM, J. Numer. Anal., Vol. 27, No. 6, 1990.

[2] E. Dahl, *Mapping and compiled communication on the Connection Machine system*, in Fifth Distributed Memory Computing Conference, 1990.

[3] M. Misra and P. Kumar, *Efficient VLSI implementation of iterative solutions to sparse linear systems*, Tech. Report 246, Institute for Robotics and Intelligent Systems, University of Southern California, 1988.

[4] B. Philippe and Y. Saad. *Solving large sparse eigenvalue problems on supercomputers*, in In Parallel and Distributed Algorithms, C. et al, ed., North-Holland, 1989.

[5] Y. Saad, *SPARSEKIT: a basic tool kit for sparse matrix computations*, Tech. Report 90 20, RIACS, NASA Ames Research Center, 1990.

[6] Y. Saad, *Practical use of polynomial preconditionings for the conjugate gradient method*, SIAM J. Sci. Stat. Comput., Vol. 6, No. 4, 1985.

[7] J. Saltz, S. Petiton, A. Rifkin, and H. Berryman, *Performance Effects of Irregular Communications Patterns on Massively Parallel Multiprocessors*, Journal of Parallel and Distributed Computing, 1991.

PARALLEL IMPLEMENTATIONS FOR SOLVING GENERALIZED EIGENVALUE PROBLEMS ARISING FROM STRUCTURAL MECHANICS

Bernard PHILIPPE and Brigitte VITAL
IRISA - Campus de Beaulieu
35042 Rennes Cedex, FRANCE

Abstract - We compare three algorithms to solve the sparse symmetric generalized eigenvalue problem on vector and parallel computers. An efficient vectorization for tridiagonalizing a banded matrix is defined. Finally, we propose a decision tree for the procedure selection.

1. INTRODUCTION

The purpose here is to solve on a parallel computer the following generalized eigenvalue problem :

$$Ax = \lambda Mx \ , \qquad (1)$$

where A and M real symmetric matrices of order n, such arise in Structural Mechanics. The assumptions imply real eigenvalues and the existence of a M-orthogonal basis of eigenvectors. The matrices A and M, which are usually obtained from a discretization, are assumed to be large and sparse. Typically, their bandwidth does not exceed $n/10$. In fact, M is either diagonal or with the same nonzero pattern as A. For non-diagonal matrices, we consider three possible storage techniques : banded matrix, profiled matrix or sparse matrix with an unstructured pattern. In all these situations, symmetric matrices are only half-stored. Since the large size of the matrices prevent the computation of the entire spectrum, the user must specify which part of the spectrum he is willing to compute. He may specify it either by limiting the interval of the eigenvalues sought or by looking for a given number of extremal eigenvalues.

The algorithms considered here are designed for parallel computers with a shared common memory such as the CRAY or ALLIANT computers. Three algorithms are developed depending on the type of storage used for the matrices. After a description of each of them, we compare their performance on some test problems and draw some conclusions about the method of choice for each situation.

All the test problems are defined on matrices which belong to the HARWELL set of test matrices. Since profile minimization may be done at a very small cost, the profile of the matrices was minimized in all cases.

2. REDUCTION TO A STANDARD EIGENVALUE PROBLEM
(A : banded matrix, M : diagonal matrix)

2.1. Analysis of the sequential algorithm

Since M is a positive definite matrix, the problem defined by Equation (1) is equivalent to the following standard eigenvalue problem :

$$Cy = \lambda y \ , \quad x = Dy \ . \qquad (2)$$

where $D = M^{-1/2}$ and $C = DAD$. Since in this section A is a banded matrix and M is diagonal, the matrix C is easily computed. It has the same band as the original matrix A. Therefore, given an interval $[a,b]$, all the eigenvalues of the original problem which belong to the interval and their corresponding eigenvectors can be computed by the following algorithm :

1. Compute the matrix C defined in Equation (2).
2. Compute a tridiagonal matrix T unitarily similar to C using Givens transformations.
3. Compute all the eigenvalues of T which belong to $[a,b]$ by iterative bisections of the interval.
4. For each eigenvalue λ, compute the corresponding eigenvector by inverse iterations (this implies the factorization of the matrix $C - \lambda I$).
5. M-reorthogonalize groups of eigenvectors corresponding to close eigenvalues by using the Modified Gram-Schmidt procedure [2].

The algorithms of Steps 2 to 4 may be found in [5]. They have been implemented in the subset of the routines which are devoted to banded matrices in the EISPACK library.

Let us consider each stage with respect to parallelism and operation counts. The half-bandwidth of A is denoted p.

Step 1 is easy to parallalize since it only involves vector operations (componentwise vector multiplications with vector length, p). Moreover, its complexity is low ($2np$ operations).

In Step 2, the tridiagonalization process involves vector operations but with vector length decreasing from p to 2 during the procedure with a $O(n^2p)$ complexity for scalar operations. The algorithm is based on a sequence of rotations which gives rise to recursions. It is essentially sequential.

A parallel algorithm for Step 3 is studied in [3]. Its efficiency is usually almost optimal but the low complexity of the step limits its impact on the overall process.

Step 4 is intrinsically parallel since the inverse iterations may be independently performed.

In Step 5, two levels of parallelism may be exploited : between different groups of vectors and within the orthogonalization of one group. The implementation is studied in [3]. To orthogonalize a group of m vectors, $2m^2n$ operations must be performed.

In conclusion, Step 2 appears to be the bottleneck for parallelizing the whole process. We propose here a new version of the algorithm which allows efficient vectorization. Theoretically, whenever vectorizing is possible, parallelizing should also be possible. However, on the computers considered here, we were not able to obtain adequate speed-ups. Hence, we restrict ourselves to the use of one vector processor.

2.2. Vectorizing the tridiagonalization of a banded matrix

The successive eliminations of entries of the banded matrix C are ordered as follows : diagonal by diagonal, starting from the outermost one. and on one diagonal, from top to botton.. Therefore, the computation is structured by a two-level loop :

```
do $q = p, p-1, ..., 2$
   do $i = 1.2, ....n - k$
      elimination of $A(i, i+q)$
```

where elimination of $A(i, i+q)$ involves a sequence of $t = \lfloor \frac{n-i}{q} \rfloor$ Givens rotations $R(k,l,\theta)$ that are successively applied to A .

$$A \ := R(k,l,\theta)^T A \ R(k,l,\theta), \ \ k = 0, \cdots, l-1. \qquad (3)$$

The sequence of rotations comes from the fact that eliminating an entry gives rise to a new nonzero entry lower down on the side of the band. This new entry will have to be eliminated later. Therefore, elimination of $A(i, i+q)$ is implemented by a two-level loop

```
do $k = 0, 1, \cdots, l-1$
   computation of $R(k,l,\theta)$
   loop for computing $A \ := A \ R(k,l,\theta)$
   loop for computing $A \ := R(k,l,\theta)^T A$
```

At the beginning of the process, the vectorized inner loops of are efficient if the band is large. However, when eliminating entries close to the main diagonal, the number of rotations is high and at the same time the vectors being manipulated are very short. In this case, a vectorizer will not yield efficient code. This problem can be rectified by commuting the inner and outer loops. This is permitted when the computation of all the rotations $R(k,l,\theta)$ may be taken out of body of the loop. This can be done by a recursion which computes the sines and cosines of the rotations [4] .

We compared these two methods of vectorization with a third combined method. For every pair (q,i), the implementation selects the best strategy between the original one (strategy A) or the strategy which is obtained by permuting the loops (strategy B). The efficiency of the approach is described in Table 1.

s = vector speed-up			
half-bandwidth (p)	7	50	100
Only (A)	s = 0.88	s = 1.51	s = 1.87
Only (B)	s = 3.20	s = 1.85	s = 1 27
switch between (A) and (B)	s = 2.93	s = 2.49	s = 2 49

Table 1: Vectorization of the tridiagonalization on Cray 2 (n=500, 1 CPU)

3. MULTISECTIONNING WITH STURM SEQUENCES (A : profiled matrix, M : diagonal or profiled matrix)

3.1. Presentation of the algorithm

When A and M are profiled matrices with the same profile, the problem defined by Equation (1) can be solved by iteratively partitioning the interval $[a, b]$ in which the eigenvalues are sought. Unlike the case in the previous section, the Sturm sequences are computed from the original matrices. The calculation implies $L - D - L^T$ factorizations of matrices $A - \lambda M$ where $\lambda \in [a, b]$. A factorization involves vector operations with a vector length equal to the half-bandwidth p. Parallelism is obtained by computing several sequences concurrently. For that purpose, we adopted a strategy similar to the strategy used in the tridiagonal situation [3], namely :

1. Isolate the eigenvalues by multisections of order *nproc*, the number of processors (a multisection is a synchronous computation of *nproc* Sturm sequences). This step results in a list of intervals enclosing single eigenvalues is obtained.
2. Extract the eigenvalues from their interval and compute the corresponding eigenvectors. Here, Sturm sequence computations are asynchronous. We selected ZEROIN as the root finding procedure [1]. Eigenvectors are computed via inverse iterations.
3. M-reorthogonalize the groups of vectors corresponding to close eigenvalues.

3.2. Parallel implementation

We illustrate the behavior of the algorithm on a problem of order $n = 817$ and where the half-bandwidth, p, of A is equal to 18 but where M is diagonal. The number of eigenpairs sought is 16. The speed-ups on ALLIANT FX/80 are displayed in Table 2. For 8 processors, the proportion of time spent in Step 1, Step 2 and Step 3 are respectively 12.3%, 87.0% and 0.6% of the total time.

nb. processors	1	2	4	8
speed-up	1	2	3.6	6.6

Table 2: Multisections with profiled matrices on an ALLIANT FX/80 (Total time on one processor : 234.6 seconds)

4. METHOD OF LANCZOS (A : unstructured sparse matrix, M : diagonal or profiled matrix ; extremal eigenvalues)

For solving Problem (1), we consider the Lanczos method with the following characteristics : block version with full reorthogonalization and dynamic restarting process. The algorithm is described in [4]. The iterative process builds an M-orthogonal system of vectors V_j and a banded matrix $T_j = V_j^T A V_j$ at every step. At every iteration, the following steps are performed :

- premultiplication of a block by A ;
- premultiplication of a block by M ;
- dense computations between blocks of vectors ;
- premultiplication of a block by M^{-1} ;
- tridiagonalization of T and computation of the eigenpairs (not at each iteration) ;
- M-reorthogonalization.

Use of blocks provides an obvious way to parallelize the algorithm by performing the transformations on the columns of a block independently. The tridiagonalization was not parallelized but only vectorized as seen in Section 2. On a problem of order $n = 1173$ where A stored by sparse diagonals and M is a profiled matrix ($p = 62$), the program was run on 4 processors of a CRAY 2. The time required to compute the 20 largest eigenpairs was 31.88 seconds (block size : $l = 20$). The corresponding speed-up was 2.9.

5. COMPARISONS AND CONCLUSION

The domain of applicability of each of the three presented algorithms is different, depending on the position of in the spectrum of the eigenvalues sought and on the matrix M. They have also different memory requirements. The highest requirement comes from the Multisections which require the storage of one profiled matrix per processor However this is also the algorithm which yelds to the highest speed-ups. To compare the methods, we present running times for two different problems in Table 3 :

- Problem (I) : $n = 1224, p = 56$, M is not diagonal.
- Problem (II) : $n = 1083, p = 62$, M is diagonal.

It is clear that, when applicable, Lanczos method is the method of choice. The decision tree for the algorithm selection is given in Figure 1.

	nb. computed eigenpairs	Reduction standard pb	Multisections	Lanczos
Pb. (I)	13	-	50 76	13.69
Pb. (II)	8	23.82	71 34	3.23

Table 3: Running times (seconds) on CRAY 2 (4 processors).

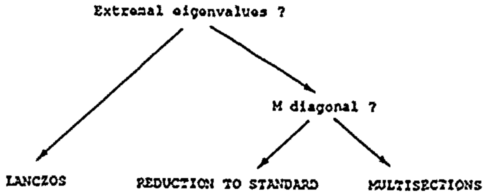


Figure 1: Decision tree for algorithm selection.

References

[1] J.C. Bus, T.J. Dekker. *Two Efficient Algorithms with Guaranteed Convergence for Finding a Zero of a Function.* ACM Trans. Math. Software, vol. 1 , pp. 330-345, 1975.

[2] Golub, G., and Van Loan, C., *Matrix computations* The Johns Hopkins University Press, Baltimore, 1983.

[3] S. S. Lo, B. Philippe, A. Sameh. *A multiprocessor algorithm for symmetric tridiagonal eigenvalue problem.* SIAM J. Stat. Scient. Comput., 8 :2, 1987.

[4] B. Vital. *Etude de quelques methodes de resolution de problemes linéaires de grande taille sur multiprocesseur.* Thesis, Université of Rennes I. 1990.

[5] J.H. Wilkinson. C. Reinsch. *Handbook for Automatic Computation.* Vol. 2, Linear Algebra, Springer Verlag, New-York, 1971.

THE LANCZOS ALGORITHM FOR THE GENERALIZED SYMMETRIC EIGENPROBLEM ON SHARED-MEMORY ARCHITECTURES*

Mark T. Jones
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

and

Merrell L. Patrick
Computer Science Department
Duke University
Durham, NC 27706

Abstract. The generalized eigenvalue problem, $Kx = \lambda Mx$, is of significant practical importance, for example, in structural engineering where it arises as the vibration and buckling problems. The paper describes the implementation of a solver based on the Lanczos algorithm, LANZ, on two shared-memory architectures, the CRAY Y-MP and Encore Multimax. Issues arising from implementing linear algebra operations on a multivector processor are examined. Portability between a multivector processor and a simple multiprocessor is discussed. Performance results from some practical problems are given and analyzed.

1. INTRODUCTION

The generalized symmetric eigenvalue problem, $Kx = \lambda Mx$, is of significant practical importance, for example, in structural engineering where it arises as the vibration and buckling problems. In the problems of interest, a few of the eigenpairs closest to some point, σ, in the eigenspectrum are sought. The matrices, K and M, are usually sparse or have a narrow bandwidth. Because eigenvalue problems arising in structural engineering are often very large, it is natural to attempt to use parallel computers to solve them. In Section 2 the parallel LANZ algorithm and its implementation on shared-memory architectures with a small to moderate number of processors is described. In Section 3 results from the implementations are given and analyzed.

2. ALGORITHM AND IMPLEMENTATION

To speed convergence to desired eigenvalues, a shift-and-invert Lanczos algorithm similar to that described in [8] is used. On sequential and vector machines, this algorithm has been observed to be superior to the subspace iteration method that is popular in engineering [9] [2]. To maintain the desired semi-orthogonality among the Lanczos vectors, a version of partial reorthogonalization [14] is used. Extended local orthogonality among the Lanczos vectors is also enforced [7] [12]. If eigenvectors are found before executing the Lanczos algorithm, an improved version of external selective orthogonalization [1] suggested in [2] is used to avoid recomputing these eigenvectors. Although the discussions in this paper assume that M is positive semi-definite, the computations remain essentially the same when M is indefinite.

The Force, a Fortran-based language for parallel programming [5], was used to implement LANZ because it is available on several shared memory architectures, thus allowing at least a superficial level of portability.

The parallel LANZ algorithm is presented in Figure 1. Its various computational components and their parallel implementations are discussed in the following subsections. Explicit global synchronization points in the algorithm are denoted by the term "SYNCHRONIZE." Other synchronization points are required by particular operations, for example inner products, and are not explicitly denoted in the algorithm. To avoid extra synchronization, each processor is responsible for computing a fixed subset of each vector computation. For example, if at step 21 processor i computes the first m elements of q_{j+1}, then at step 22, processor i would compute the contribution of the first m elements to the inner product, thus avoiding a synchronization between steps 21 and 22. In these discussions p represents the number of processors, n represents the order of the matrices, b represents the block size in a block algorithm, and j represents the current Lanczos step.

```
0)  $q_0 = p_0 = 0$
1)  Choose an initial vector, $guess$
2)  $p_1 = M guess$
3)  Orthogonalize
4)  SYNCHRONIZE
5)  $p_1 = M guess$
6)  SYNCHRONIZE
7)  $q_1 = (K - \sigma M)^{-1} p_1$
8)  (factorization occurs here)
9)  SYNCHRONIZE
10) $p_1 = M q_1$
11) $\beta_1 = (p_1^T q_1)^{\frac{1}{2}}$
12) Orthogonalize
13) $q_1 = q_1 / \beta_1$
14) $p_1 = p_1 / \beta_1$
15) For $j = 1, \ldots$
16)     $(K - \sigma M) q_{j+1} = p_j$
17)     (only matrix solution here)
18)     SYNCHRONIZE
19)     rnorm=$\| q_{j+1} \|$
20)     (if external orthogonalization)
21)     $\gamma = p_{j-1}^T q_{j+1}$
22)     $q_{j+1} = q_{j+1} - \gamma q_{j-1}$
23)     $\hat{\alpha} = p_j^T q_{j+1}$
24)     $q_{j+1} = q_{j+1} - \hat{\alpha} q_j$
25)     $\gamma = p_{j-1}^T q_{j+1}$
26)     $q_{j+1} = q_{j+1} - \gamma q_{j-1}$
27)     $\alpha_j = p_j^T q_{j+1}$
28)     $q_{j+1} = q_{j+1} - \alpha_j q_j$
29)     SYNCHRONIZE
30)     $p_{j+1} = M q_{j+1}$
31)     $\alpha_j = \alpha_j + \hat{\alpha}$
32)     $\beta_{j+1} = (p_{j+1}^T q_{j+1})^{\frac{1}{2}}$
33)     Calculate eigenvalues of $T_j$
34)     Count the converged eigenvalues
35)     Orthogonalize
36)     $q_{j+1} = q_{j+1} / \beta_{j+1}$
37)     (requires use of critical sections)
38)     $p_{j+1} = p_{j+1} / \beta_{j+1}$
39) End of Loop
40) compute ritz vectors
```

Fig. 1. *Parallel shift-and-invert Lanczos algorithm*

2.1. Factorization

Factorization takes place only once during the algorithm, at step 7 Because the matrices, K and M, are sparse (or have been reordered to have a narrow bandwidth), the parallel implementation of direct factorization and solution methods must be carefully considered. In this paper, only the case in which the matrices have been reordered to a narrow bandwidth, β, will be considered. However, the limitations on parallelism in factorization and forward/backward matrix solution that are imposed by a narrow bandwidth are similar to those imposed by sparse matrices.

Two situations may exist when factoring $(K - \sigma M)$. (1) $(K - \sigma M)$ is known to be positive definite, and therefore it is desirable to use either Cholesky factorization or LDL^T decomposition, or (2) $(K \sigma M)$ may be indefinite, and therefore a factorization algorithm with pivoting is necessary. In the first case, a block factorization and

*This research was supported by the National Aeronautics and Space Administration under NASA contract nos. NAS1-18107 and NAS1-18605 while the authors were in residence at ICASE. Additional support was provided by NASA grant no. NAG-1-466. The first author also received support from the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. The authors also received time on a CRAY Y-MP at the North Carolina Supercomputing Center.

solution subroutine described in [13] has been parallelized for use in LANZ. In the second case, a block algorithm for banded matrices based on Bunch-Kaufman factorization is used [3].

LANZ was initially written for vector architectures, and therefore careful attention has been paid to achieving good vectorization. With small-to-moderate vector lengths, it is desirable to perform *saxpy* operations[1] as opposed to inner products, as well as to compute more than one *saxpy* operation at a time.[2] On multivector processors, however, good vectorization is often at odds with parallelization. In the factorization algorithms, this conflict between vectorization and parallelization occurs in the computation of the pivot column(s): the pivot columns(s), vectors of length β, must be split into vectors of length β/p for each processor to compute. On the CRAY Y-MP the benefit of parallel computation of the pivot column is outweighed by the resulting inefficient short vector operations and the cost of the added synchronization; therefore, this computation is not parallelized. However, on a simple multiprocessor such as the Multimax, this conflict does not occur, and the computation of the pivot column is parallelized. The dominant part of the calculation is the updating of the uneliminated nonzeroes by using the pivot columns: the updating is implemented by distributing $\frac{\beta-b}{p}$ extended *saxpy*'s to each of the processors to compute. The extended *saxpy*'s parallelize well because there is sufficient work for each processor, and the vector lengths are unaffected by parallelization.

2.2. Matrix Solution

Forward and backward matrix solution is required at steps 7 and 16. The conflict between vectorization and parallelism is much worse in these operations. This discussion will be limited to the forward and backward solution algorithms that take place after a Bunch-Kaufman factorization in which the block sizes vary and are selected according to numerical criteria rather than the number of processors.[3] The following discussion will assume that the lower triangular factor, L, resulting from the Bunch-Kaufman algorithm has been stored by row.[4] Because of the order in which pivots are performed, a *saxpy*-based algorithm for the forward solution must be used, and an inner product algorithm for the backward solution must be used.

The time-consuming portion of the block forward solution algorithm is the b β-length *saxpy* operations that can be combined into a single extended *saxpy* operation. The only practical way to parallelize this operation is to split the vector into p shorter vectors. This approach, of course, significantly reduces the efficiency of the vector operations.

The time-consuming portion of the block backward solution algorithm is the computation of b β-length inner products. Two types of parallelism are available here. (1) two or more processors can cooperate to compute a single inner product, and (2) individual inner products can be computed independently. Even though both methods are used, the algorithm is still inefficient because inner products are not as fast as *saxpy*'s, the parallel computation of a short inner product is adversely affected by synchronization delays, and the block size may not be evenly divisible by p, and therefore a load imbalance may result.

The considerations regarding efficiency of vector operations are not a concern when implementing this algorithm on the Encore, and therefore better parallel speedup from the forward and backward solution algorithms can be expected than on the CRAY Y-MP. The ratio of computation to synchronization, however, is still much worse than for factorization, and good speedup cannot be expected.

2.3. Other Computations

The parallelization of other computations in the algorithm, including sparse matrix multiplication, solution of the small tridiagonal eigenproblem, Ritz vector computation, and orthogonalization and discussed more fully in [4].

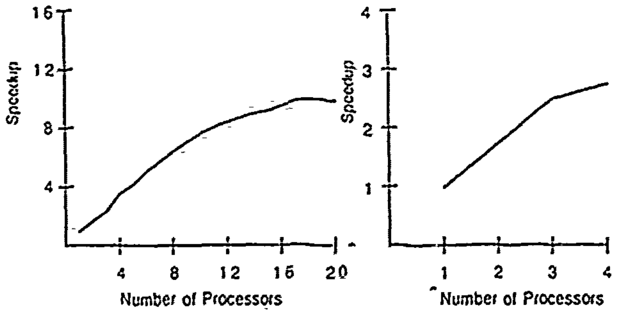


FIG. 2. *Speedup curves*

3. PERFORMANCE RESULTS AND ANALYSIS

As a demonstration of its performance, LANZ was run on a medium size eigenproblem from structural engineering[5] where the ten lowest eigenpairs were found in 22 steps on a four processor CRAY Y-MP A smaller problem[6] was run on a twenty processor Encore Multimax in which the ten lowest eigenpairs were found in 22 steps. It is clear from the speedup curves in Figure 2 that a speedup plateau occurs. The main cause of this plateau is the poor speedup realized in the forward and backward matrix solution algorithms. The problem caused by the matrix solution algorithms is exacerbated as the number of Lanczos steps increases, because each Lanczos step requires another forward/backward matrix solution, taking more and more time as compared to factorization, which speeds up well. This plateau occurs later on the Encore than the CRAY because the Encore does not have to contend with the conflict between vectorization and parallelization: vector lengths decreasing as the number of processors increases. However, both implementations suffer from the poor ratio of computation to synchronization in the forward and backward matrix solution algorithms.

If problems with larger bandwidths were used, better speedup from these algorithms could, of course, be expected. It has been the authors' experience, however, that if the bandwidth arising from a structural engineering problem is large, then most likely many zeroes exist inside the band, and therefore sparse methods are best used.

[1] The *saxpy* operation is defined as $w = \alpha z + y$, where w, y, and z are vectors and α is a scalar.

[2] Performing more than one *saxpy* at a time, called an *extended saxpy* in this paper, is defined as $w = y + \sum_{i=1}^{l} \alpha_i z_i$ and is often implemented via loop unrolling. This type of operation reduces the ratio of memory references to computations.

[3] The situation is slightly better for the positive definite case in which the block sizes can be selected based on the number of processors rather than according to numerical criteria.

[4] If it were stored by column, the same limitations would apply, but the discussion for the forward solution would be applicable to backward solution and vice versa.

[5] Finding the vibration modes and mode shapes of the finite element model of a circular cylindrical shell [15]. In this problem $n = 12054$ and the average semi-bandwidth is 394.

[6] Finding the five lowest buckling modes and mode shapes of the finite element model of an I stiffened panel. In this problem $n = 4474$ and the average semi-bandwidth was 207.

4. CONCLUDING REMARKS

A parallel Lanczos algorithm for finding a few of the eigenpairs around a point in the eigenspectrum was described. Differences in the implementation of the algorithm on a multivector processor and on a multiprocessor were described. The algorithm was shown to perform reasonably well on a moderate number of processors. A performance bottleneck which prevents efficient utilization of a large number of processors was identified.

Several possible modifications to the LANZ algorithm can be used to improve its parallel performance. The use of dynamic shifting [1] to improve parallelism by reducing the number of forward and backward matrix solutions was investigated in [2] and was found to be successful when the eigenvalue distribution was difficult. The use of groups of processors executing the LANZ algorithm independently at different shifts was investigated in [2] and was found to be successful when many eigenpairs are being sought. Block Lanczos holds some promise because it allows several forward and backward matrix solutions to occur simultaneously [1]. The improvement in performance resulting from block Lanczos will depend on how many Lanczos steps are eliminated and what block size can be effectively used. Unfortunately, *s*-step Lanczos methods [6] will not alleviate the bottleneck imposed by forward and back solutions and, therefore, will not have a significant effect on performance.

Another avenue for improving parallel performance is the use of iterative methods to solve $(K - \sigma M)x = y$ rather than direct methods. However, it has been the authors' experience that $(K - \sigma M)$ is often poorly conditioned and, therefore, is difficult to solve by iterative methods.

REFERENCES

[1] R. G. Grimes, J. G. Lewis, and H. D. Simon, *The Implementation of a Block Lanczos Algorithm with Reorthogonalization Methods*, ETA-TR 91, Boeing Computer Services, Seattle, Washington, May 1988.

[2] M. T. Jones, *The Use of Lanczos' Method to Solve the Generalized Eigenproblem*, PhD thesis, Department of Computer Science, Duke University, 1990.

[3] M. T. Jones and M. L. Patrick, *Factoring Symmetric Indefinite Matrices on High-Performance Architectures*, Technical Report 90-8, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Va., 1990.

[4] ——, *The Lanczos Algorithm for the Generalized Symmetric Eigenproblem on Shared-Memory Architectures*, Preprint MCS-P182-1090, MCS Division, Argonne National Laboratory, Argonne, Il., 1990.

[5] H. Jordan, *The Force*, Computer Systems Design Group, University of Colorado, 1987.

[6] S. Kim and A. Chronopoulos, *A Class of Lanczos-Like Algorithms Implemented on Parallel Computers*, Computer Science Department 89-49, University of Minnesota, July 1989.

[7] J. G. Lewis, *Algorithms for Sparse Matrix Eigenvalue Problems*, PhD thesis, Department of Computer Science, Stanford University, 1977

[8] B. Nour-Omid, B. N. Parlett, T. Ericsson, and P. S. Jensen, *How to Implement the Spectral Transformation*, Mathematics of Computation, 48 (1987), pp. 663–673.

[9] B. Nour-Omid, B. N. Parlett, and R. L. Taylor, *Lanczos Versus Subspace Iteration For Solution of Eigenvalue Problems*, International Journal for Numerical Methods in Engineering, 19 (1983), pp. 859–871.

[10] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J., 1980.

[11] B. N. Parlett and B. Nour-Omid, *The Use of a Refined Error Bound When Updating Eigenvalues of Tridiagonals*, Linear Algebra and its Applications, 68 (1985), pp. 179–219.

[12] B. N. Parlett, B. Nour-Omid, and Z. A. Liu, *How to Maintain Semi-Orthogonality Among Lanczos Vectors*, PAM-420, Center for Pure and Applied Mathematics, University of California, Berkeley, July 1988.

[13] E. Poole, *The Solution of Linear Systems of Equations with a Structural Analysis Code on the NAS Cray 2*, Tech. Rep. CR-4159, CSM Branch, NASA Langley Research Center, Hampton, Va., 1988.

[14] H. D. Simon, *The Lanczos Algorithm With Partial Reorthogonalization*, Mathematics of Computation, 42 (1984), pp. 115–142.

[15] C. B. Stewart, *The Computational Structural Mechanics Testbed Procedure Manual*, Computational Structural Mechanics Branch, NASA Langley Research Center, 1989.

A DIVIDE AND CONQUER APPROACH TO THE NONSYMMETRIC EIGENVALUE PROBLEM

ELIZABETH R. JESSUP
Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Building 6012
Oak Ridge, TN 87831-6367 USA

Abstract Serial computation combined with high communication costs on distributed-memory multiprocessors make parallel implementations of the QR method for the nonsymmetric eigenvalue problem inefficient. This paper introduces an alternative algorithm for the nonsymmetric tridiagonal eigenvalue problem based on rank two tearing and updating of the matrix. The parallelism of this divide and conquer approach stems from independent solution of the updating problems.

I. Introduction

The eigenvalues and eigenvectors of a nonsymmetric matrix A have traditionally been computed by first reducing A to Hessenberg form H and then computing the eigendecomposition of H by the QR method. The serial nature of the QR method combined with the high cost of data transfer on distributed-memory multiprocessors has made parallel implementations of this approach inefficient [7]. In this paper, we outline an alternative algorithm for the nonsymmetric eigenvalue problem. The algorithm uses a divide and conquer technique and follows from methods that have performed well both serially and in parallel for the symmetric tridiagonal [4] and unitary [2] eigenvalue problems and for the bidiagonal singular value problem [10].

The method is presented here for nonsymmetric tridiagonal matrices. Matrices of this form arise directly from certain applications [1] and from reduction of general matrices to tridiagonal form [5]. We expect ultimately to extend our divide and conquer technique to general matrices.

Throughout this paper, unless otherwise specified, capital Greek and Roman letters represent matrices, lower case Roman letters represent column vectors, and lower case Greek letters represent scalars. A superscript T denotes transpose. The vector e_j is the j-th "canonical vector" with all elements equal to zero except the j-th which equals 1.

II. The Algorithm

Let T be the following $n \times n$ real, tridiagonal, irreducible, nondefective nonsymmetric matrix

$$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \gamma_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \gamma_n & \alpha_n \end{pmatrix}. \qquad (1)$$

By splitting off two superdiagonal elements β_m and β_{m+1} and the corresponding subdiagonal elements γ_m and γ_{m+1}, we can write the matrix T in terms of the tridiagonal nonsymmetric submatrices T_1 and T_2:

$$T = \left(\begin{array}{c|c|c} T_1 & & \\ \hline & \alpha_m & \\ \hline & & T_2 \end{array}\right) + \left(\begin{array}{c|c|c} & \beta_m & \\ \hline \gamma_m & & \beta_{m+1} \\ \hline & \gamma_{m+1} & \end{array}\right). \qquad (2)$$

If T_1 and T_2 are nondefective, we can compute the eigendecompositions $T_1 = X_1 D_1 X_1^{-1}$ and $T_2 = X_2 D_2 X_2^{-1}$. Substituting these decompositions and abbreviating $\alpha = \alpha_m$ gives the matrix product

$$\begin{aligned} T &= \left(\begin{array}{c|c|c} X_1 D_1 X_1^{-1} & & \\ \hline & \alpha & \\ \hline & & X_2 D_2 X_2^{-1} \end{array}\right) + \left(\begin{array}{c|c|c} & \beta_m & \\ \hline \gamma_m & & \beta_{m+1} \\ \hline & \gamma_{m+1} & \end{array}\right) \\ &= \hat{X} \left[\left(\begin{array}{c|c|c} D_1 & & \\ \hline & \alpha & \\ \hline & & D_2 \end{array}\right) + \left(\begin{array}{c|c|c} & v_1 & \\ \hline h_1^T & & h_2^T \\ \hline & v_2 & \end{array}\right) \right] \hat{X}^{-1}, \end{aligned} \qquad (3)$$

where

$$\hat{X} = \left(\begin{array}{c|c|c} X_1 & & \\ \hline & 1 & \\ \hline & & X_2 \end{array}\right),$$

$v_1 = \beta_m X_1^{-1} e_m$, $v_2 = \gamma_{m+1} X_2^{-1} e_1$, $h_1 = \gamma_m X_1^T e_m$, and $h_2 = \beta_{m+1} X_2^T e_1$ for canonical vectors e_1 and e_m of appropriate length. (The algorithm can be reformulated to account for defective matrices by replacing the eigendecompositions $T_1 = X_1 D_1 X_1^{-1}$ and $T_2 = X_2 D_2 X_2^{-1}$ with ones including the rank deficient left and right eigenvector matrices $T_1 = X_1 D_1 Y_1$ and $T_2 = X_2 D_2 Y_2$.)

We permute the elements of equation (3) to form

$$T = X \left[\left(\begin{array}{cc|c} D_1 & & \\ & D_2 & \\ \hline & & \alpha \end{array}\right) + \left(\begin{array}{cc|c} & & v_1 \\ & & v_2 \\ \hline h_1^T & h_2^T & 0 \end{array}\right) \right] X^{-1} \qquad (4)$$

with

$$X = \left(\begin{array}{cc|c} X_1^{-1} & & \\ & X_2^{-1} & \\ \hline & & 1 \end{array}\right)$$

and rewrite the interior matrix of equation (4) as

$$M = \left(\begin{array}{cc|c} D_1 & & \\ & D_2 & \\ \hline & & \alpha \end{array}\right) + \left(\begin{array}{cc|c} & & v_1 \\ & & v_2 \\ \hline h_1^T & h_2^T & 0 \end{array}\right) = \begin{pmatrix} D & v \\ h^T & \alpha \end{pmatrix}. \qquad (5)$$

The matrix M is the sum of a diagonal matrix and a rank two nonsymmetric matrix. The eigenvalues of the matrix M are the eigenvalues of T. The left and right eigenvectors of M postmultiplied by X^T and X^{-1}, respectively, are the left and right eigenvectors of T.

The procedure for computing the eigenvalues and eigenvectors of M follows basic steps similar to those for the eigendecomposition of a diagonal plus symmetric rank one matrix developed in [3,8], but because M is nonsymmetric, the details differ in several important ways. Let $\delta_1, \ldots, \delta_n$ denote the diagonal elements of D. If no eigenvalue λ of M equals a diagonal element δ_j, then the eigenvalues of M are the zeros of the rational equation

$$g(\lambda) = (\lambda - \alpha) + \sum_{i=1}^{n-1} \frac{(e_i{}^T h)(e_i{}^T v)}{\delta_i - \lambda} = 0 \qquad (6)$$

which may be solved by a complex rootfinder such as Muller's method [11]. Because they are also the eigenvalues of a submatrix of the real tridiagonal matrix T, the complex eigenvalues of M occur in conjugate pairs.

Once the eigenvalues have been computed, it is a straightforward matter to compute the eigenvectors using the expression for the matrix M. Let λ be a computed eigenvalue of M. One expression for the corresponding right eigenvector $q^T = (\hat{q}^T, \quad \xi)$ comes from the relation

$$\begin{pmatrix} D & v \\ h^T & \alpha \end{pmatrix} \begin{pmatrix} \hat{q} \\ \xi \end{pmatrix} = \begin{pmatrix} D\hat{q} + \xi v \\ h^T \hat{q} + \alpha \xi \end{pmatrix} = \lambda \begin{pmatrix} \hat{q} \\ \xi \end{pmatrix}.$$

Specifically, if the matrix $(D - \lambda)$ is nonsingular, the right eigenvector is given by

$$q = \begin{pmatrix} \hat{q} \\ \xi \end{pmatrix} = \begin{pmatrix} -(D - \lambda)^{-1} v \\ 1 \end{pmatrix} \xi, \qquad (7)$$

where ξ is selected to make $q^T q = 1$. A similar expression can be derived for the corresponding left-singular vector.

To this point, the derivation of the divide and conquer method for the nonsymmetric eigenvalue problem requires that the matrix $(D - \lambda I)$ be nonsingular. For this to be the case, no eigenvalue of M may equal a diagonal element of D. Equivalently, no element of v can be zero, no element of h can be zero, and no diagonal elements of D may be equal. We now show how to deflate the problem in exact arithmetic so that none of these situations arises. The deflation rules not only produce a correct form for the matrix M but also reduce the amount of computation needed for solving its eigensystem.

First, if the jth element of v is zero, then $e_j^T M = \lambda_j e_j^T$ where e_j is the jth canonical vector of length n. Thus, λ_j is an eigenvalue of M and e_j is its corresponding left eigenvector. Likewise, if the jth element of h is zero, then $Me_j = \lambda_j e_j$, so that (λ, e_j) is a right eigenpair of M. Rows and columns of the matrix M having all zero off-diagonal elements can be removed from the matrix and the eigenvalue problem deflated.

When $\delta_i = \delta_j$, we can apply unitary similarity transformations to reduce $h_i = e_i^T h$ to zero and deflate the problem. Let

$$\tau^2 = |h_j|^2 + |h_i|^2, \quad c = \frac{h_j}{\tau}, \quad s = \frac{h_i}{\tau},$$

then the matrix is transformed in the following way

$$\begin{pmatrix} \bar{c} & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_i & 0 \\ 0 & \delta_j \\ h_i & h_j \end{pmatrix} \begin{pmatrix} c & \bar{s} \\ -s & \bar{c} \end{pmatrix} = \begin{pmatrix} \delta_i & 0 \\ 0 & \delta_j \\ 0 & \tau \end{pmatrix}.$$

After transformation, δ_i is an eigenvalue of M with right eigenvector e_i. Note that the zero structure of the deflated M guarantees that the $n \times n$ matrix $M - \lambda I$ has rank at least $n - 1$. Thus, M has distinct eigenvalues and a complete set of eigenvectors.

As in the symmetric case, we expect the divide and conquer method to be a good parallel method. The rank two tearing of equation (2) can be applied recursively to tridiagonal submatrices T_1 and T_2. Parallelism arises both at the subproblem level (computing $T_1 = X_1 D_1 X_1^{-1}$ and $T_2 = X_2 D_2 X_2^{-1}$ in parallel) and at the rootfinding level. Both levels can be exploited in shared-memory implementations of divide and conquer algorithms [4,10], but no implementation on a distributed-memory multiprocessor has attempted to parallelize the rootfinding tasks [9]. As in the symmetric case, the nonsymmetric divide and conquer algorithm can be pipelined with reduction of a general matrix to tridiagonal form by the algorithm in [6]. As soon as a submatrix has been formed, its eigendecomposition is computed. Once additional subproblems have been solved, rank one updating can begin.

References

[1] G. Ammar, D. Cheng, W. Dayawansa, and C. Martin, *Identification of linear systems by Prony's method*, in Robust Control of Linear Systems and Nonlinear Control, 1990, pp. 483–488.

[2] G. Ammar, L. Reichel, and D. Sorensen, *An implementation of a divide and conquer method for the unitary eigenproblem*. To appear in ACM TOMS.

[3] J. Bunch, C. Nielsen, and D. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[4] J. Dongarra and D. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s139–s154.

[5] G. Geist, *Reduction of a general matrix to tridiagonal form*, Tech. Report ORNL/TM-10991, Oak Ridge National Laboratory, 1989.

[6] G. Geist, A.Lu, and E. Wachspress, *Stabilized reduction of an arbitrary matrix to tridiagonal form*, Tech. Report ORNL/TM 11089, Oak Ridge National Laboratory, 1989.

[7] G. Geist and G. Davis, *Finding eigenvalues and eigenvectors of unsymmetric matrices using a distributed-memory multiprocessor*, Tech. Report ORNL/TM-10938, Oak Ridge National Laboratory, 1988.

[8] G. Golub, *Some modified matrix eigenvalue problems*, SIAM Review, 15 (1973), pp. 318–34.

[9] I. Ipsen and E. Jessup, *Solving the symmetric tridiagonal eigenvalue problem on the hypercube*, SIAM J. Sci. Stat. Comput., Vol. 11, No. 2, (1990), pp. 203–229.

[10] E. Jessup and D. Sorensen, *A parallel algorithm for computing the singular value decomposition of a matrix*, Technical Report ANL/MCS-TM-102, Argonne National Laboratory, 1987.

[11] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

A PARALLEL VARIANT OF GMRES(m)

ERIC DE STURLER
Delft University of Technology
Faculty of Technical Mathematics and Informatics
P.O.box 356, NL-2600 AJ Delft, The Netherlands

Abstract: In the usual implementation of GMRES(m) [3] the computationally most expensive part is the Modified Gram-Schmidt process (MGS). It is obvious, that the MGS process is not well parallelizable on distributed memory multiprocessors, since the inner products act as synchronization points and thus require communication that cannot be overlapped. Furthermore, as all orthogonalizations must be done sequentially, MGS generates a large number of short messages, which is relatively expensive. Especially on large processor grids the time spent in communication in the MGS process may be significant. For this reason a variant of the usual GMRES(m) algorithm is considered, called modGMRES(m), which first generates the vectors that span the Krylov space and then combines the MGS steps for a group of vectors. It is shown, on real world problems, that the modGMRES(m) method can yield a considerable gain in time per iteration. Numerical experience suggests that the total number of iterations remains about the same as for GMRES(m).

Introduction

As described in [2] at SHELL's KSEPL the reservoir simulator Bosim has been parallelized on a Meiko Computing Surface, a transputer based parallel computer. The parallelization is based on a 2-D domain decomposition, where the reservoir is divided among a 2-D grid of processors, in addition there is a master processor, which handles the initialization and input/output. In Parallel Bosim the largest part of the computation is done in Fortran. On each processor, the Fortran program runs in parallel with an Occam process. The Occam processes on different processors form a harness that takes care of the communication. When Fortran on a processor has to communicate, it sends the data to the local Occam process and then continues until its next communication. Meanwhile, the Occam processes concurrently take care of the communication and see to it that the data is returned to Fortran on the receiving processor(s). In this way communication and computation can be overlapped. Within this Parallel Bosim package the GMRES(m) and modGMRES(m) method were implemented and compared. In Bosim the convergence in the linear solver is checked by a separate routine, which needs the residual vector. Therefore the convergence in the linear solver is only checked after each complete (mod)GMRES(m) cycle.

The modGMRES(m) method

Let $Ax = b$ be the preconditioned system, and let $r = b - Ax$ be the residual. Let v_1 be the normalized residual, then from v_1 a suitable set of vectors $\hat{v}_2, \ldots, \hat{v}_{m+1}$, which span the Krylov space, is generated (see below). Then in the MGS process these vectors are orthogonalized. Because the Krylov space is generated in a different way, the Hessenberg matrix $H_{m+1} = V^T AV$ must be computed from the coefficients of the MGS process, where V is the matrix $[v_1, v_2, \ldots, v_{m+1}]$. The rest of the method is analogous to the GMRES(m) method. See however also [1].

In the MGS process the vectors $v_1, \hat{v}_2, \ldots, \hat{v}_{m+1}$ are orthogonalized in the following steps:

1. orthogonalize $\hat{v}_2, \ldots, \hat{v}_{m+1}$ on v_1, normalize $\hat{v}_2$, which gives v_2
2. orthogonalize $\hat{v}_3, \ldots, \hat{v}_{m+1}$ on v_2, normalize $\hat{v}_3$, which gives v_3

$\vdots$

Each step can be done blockwise, because the orthogonalizations in one step are all independent. As the innerproducts computed on each processor are strictly local, these local coefficients must be accumulated over the processor grid to compute the global coefficients. This is done blockwise by the routines accs(array,len), which sends the array to Occam and returns so that Fortran can continue, while Occam performs the actual accumulation in parallel, and accr(array,len), which receives the accumulated values from Occam. The MGS process is implemented as follows (locally on each processor):

```
do $i = 1, m - 1$
    $nbl = max(1, (m - i)/2)$
    do $k = i, i + nbl - 1$
        $h(k, i) = v_i^T v_{k+1}$
    accs($h(i, i), nbl$)
    do $k = i + nbl, m - 1$                 [concurrent with]
        $h(k, i) = v_i^T v_{k+1}$           [ accumulation ]
    accr($h(i, i), nbl$)
    $v_{i+1} = v_{i+1} - h(i, i) * v_i$
    $h(m, i) = v_{i+1}^T v_{i+1}$
    accs($h(i + nbl, i), m - \imath - nbl + 1$)
    do $k = i + 1, i + nbl - 1$             [concurrent with]
        $v_{k+1} = v_{k+1} - h(k, i) * v_i$   [ accumulation ]
    accr($h(i + nbl, i), m - i - nbl + 1$)
    do $k = i + nbl, m - 1$
        $v_{k+1} = v_{k+1} - h(k, i) * v_\imath$
    $h(i, i + 1) = sqrt(h(m, i))$
    $v_{i+1} = h(i, i + 1)^{-1} * v_{i+1}$
```

In this implementation communication is mostly overlapped and also time is saved by combining small messages, corresponding to one group of orthogonalizations, in one large message, which saves startup times and Fortran-Occam communication.

The generation of the vectors, that span the Krylov space, can be handled in two ways. If the condition number of the preconditioned system is sufficiently small and m is also relatively small (e.g. 10 or 20), the vectors can be generated as v_1, Av_1, A^2v_1, $\ldots$, $A^m v_1$, where A is the preconditioned matrix; this will be referred to as version 1. However for larger m and/or a preconditioned matrix A with a large condition number, the matrix $[v_1, Av_1, \ldots, A^m v_1]$ will be so poorly conditioned that orthogonalization with the MGS algorithm will not produce a set of sufficiently orthogonal v_i and consequently will result in an inaccurate representation of H_{m+1}. Therefore the v_i might be generated as follows:

```
do $i = 1, m$
    $\hat{v}_{i+1} = \hat{v}_i - d_i A\hat{v}_i$,  ( $\hat{v}_1 = v_1$ )
```

where the d_i are parameters, which should be chosen in such a way that the condition number of the matrix $[v_1, \hat{v}_2, \ldots, \hat{v}_{m+1}]$ is sufficiently small. This will be referred to as version 2. The computation of the d_i can be based, for example, on the eigenvalues of H_{m+1}. The exact computation of these parameters is however outside the scope of this paper.

Computational and Communication Costs of GMRES(m) and modGMRES(m)

The total computational costs will be expressed in terms of the timings of the main computational kernels. The computational costs for GMRES(m) on each processor are given by:

$$\begin{aligned} C_{gmres}^{cp} &= \frac{1}{2}(m^2 + 3m)T_{ddot} + \frac{1}{2}(m^2 + 3m)T_{daxpy} + \\ &\quad (m + 1)T_{dscal} + (m + 1)T_{mat} + \\ &\quad (m + 1)T_{prec} + T_{lincon} \qquad (1) \end{aligned}$$

The computational costs for modGMRES(m) version 1 on each processor are given by:

$$
\begin{aligned}
C^{cp}_{mgm,1} &= \frac{1}{2}(m^2+3m)T_{ddot} + \frac{1}{2}(m^2+3m)T_{daxpy} + \\
& (m+1)T_{dscal} + (m+1)T_{mat} + \\
& (m+1)T_{prec} + T_{lincon} + T_{hess}(m) \qquad (2)
\end{aligned}
$$

and for version 2 by:

$$
\begin{aligned}
C^{cp}_{mgm,2} &= \frac{1}{2}(m^2+3m)T_{ddot} + \frac{1}{2}(m^2+5m)T_{daxpy} + \\
& (m+1)T_{dscal} + (m+1)T_{mat} + \\
& (m+1)T_{prec} + T_{lincon} + T_{hess}(m) \qquad (3)
\end{aligned}
$$

where lincon is the routine that checks whether convergence is reached and hess is the routine that computes H_{m+1} in modGMRES(m). The communication costs for GMRES(m) are given by:

$$C^{cm}_{gmres} = \frac{1}{2}(m^2+3m)T_{accsum} \qquad (4)$$

where accsum is a Fortran routine that sends one number to Occam, waits for Occam to accumulate the values over the processor grid and receives back the global result. The communication costs for modGMRES(m) are given by:

$$C^{cm}_{mgm} = 2mT_{accs} + 2mT_{accr} + T_{ovh}(m) + T_{noc} \qquad (5)$$

$$T_{ovh}(m) = 2mT_c + \frac{1}{2}m^2T_{cpn} \qquad (6)$$

where T_{ovh} indicates the time overhead measured in the daxpy's and ddot's which overlap the accumulation, T_c is some constant time and T_{cpn} gives a time increase per number. T_{noc} indicates the cost of non-overlapped communication/accumulation.

Results

The GMRES(m) and the modGMRES(m)-method were compared simulating a real reservoir for a large number of iterations. As the simulation had to be done on a fairly small machine (1 master and 24 gridnodes), whereas the modGMRES(m) method will only be really advantageous on a relatively large processor grid, involving 100 or more processors, a small reservoir was simulated on a 1D (line) processor grid. This gives 24 communication steps for accumulating a distributed value over the grid, which compares to a processor grid of some 150 processors. The number of unknowns in this problem was 2138, which gave a maximum of 105 unknowns on a processor. With an average of some 90 unknowns per processor and 150 processors this compares to a reservoir with about 13500 (active) gridblocks, which is a medium scale reservoir model. This produced the following average timings on the busiest processor:

computational costs	time (ms)	communication costs	time (ms)
T_{daxpy}	0.30594	T_{accsum}	0.62041
T_{ddot}	0.24707	T_{accs}	0.16000
T_{dscal}	0.12354	T_{accr}	0.06400
T_{mat}	2.8251	T_c	0.35625
T_{prec}	2.4320	T_{cpn}	0.16293
T_{lincon}	3.0336	T_{noc}	6.9964
$T_{hess}(10)$	1.2060		
$T_{hess}(20)$	7.4160		
$T_{hess}(50)$	96.046		

Inserting these timings in (1) and (4) for GMRES(m), in (2), (5) and (6) for modGMRES(m) version 1, for m = 10 or 20, and in (3), (5) and (6) for modGMRES(m) version 2, for m = 50, leads to the following tables:

m	time (ms)		time diff.
	modGMRES(m)	GMRES(m)	(%)
10	126.12	138.49	10
20	313.43	385.91	23
50	1390.2	1832.3	32

m	efficiency (%)	
	modGMRES(m)	GMRES(m)
10	66	60
20	66	54
50	63	47

These theoretical efficiency figures are based upon the model described above, with 13500 grid blocks, a processor grid of 150 processors and one master processor. The number of communication steps for accumulating a distributed value is 24 and the maximum number of blocks on a processor is 105. Furthermore communication costs, other than in the global accumulation of distributed values, are neglegible. For GMRES(10), GMRES(20), modGMRES(10) and modGMRES(20) the overall timings were also determined experimentally on the busiest processor, which leads to the following table, containing the average time per iteration:

m	time (ms)		time diff.
	modGMRES(m)	GMRES(m)	%
10	126.09	137.13	9
20	314.58	383.94	22

From these tables it is obvious that the modGMRES(m) method can yield a substantial improvement in time per iteration. Furthermore, for these simulations, there was no difference in the total number of iterations between GMRES(m) and modGMRES(m). With respect to the (theoretical) parallel efficiency, it can be seen that, for increasing m, the decrease in efficiency of modGMRES(m) is much less than for GMRES(m) This is explained by the fact, that the communication costs increase exponentially with m, and these are much higher for GMRES(m) than for modGMRES(m).

References

[1] Chronopoulos A.T., Gear C.W., s-step iterative methods for symmetric linear systems, J. Comp. Appl. Math. 25 (1989) 153-168 North-Holland

[2] van Daalen D.T., Hoogerbrugge P.J., Meijerink J.A., Zeestraten R.J.A., Publication 924, Koninklijke/Shell Exploratie en Produktie Laboratorium Rijswijk, The Netherlands, Shell Research B.V. 1989

[3] Saad Y., Schultz M.H., GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, SIAM J.Sci.Stat.Comp 7 (no. 3) July 1986

PERFORMANCE OF ITERATIVE METHODS FOR DISTRIBUTED MEMORY MACHINES*

D.C. Marinescu, J.R. Rice and E.A. Vavalis
Purdue University
Computer Science Department
West Lafayette, IN 47907, U.S.A.

Abstract: We consider the performance of the *Jacobi SI* iterative method applied to linear systems arising from partial differential equation problems. The E/T methodology is applied to experimental data from a 128 processor NCUBE 1. The observed performance of the implementation is seen to be poor, reasons for this are presented and remedies suggested.

1. INTRODUCTION

We describe the applications of the Event Thread of Control (E/T) methodology for parallel performance evaluation to iterative methods implemented on an NCUBE 1 (a distributed memory machine). The E/T methodology is to collect traces of events and from their general behavior infer certain aspects of program performance, see [2]. Several kinds of events are traced, of interest here are *Read* and *Write* events as they relate to the communication and synchronization delays that occur in parallel iterative methods.

The key item in the E/T methodology is the *characteristic function* g defined by $E = g(P)$ where E is the number of events and P is the number of threads or processors. The methodology uses assumptions of conservation and monoticity of work to derive various relationships among performance measures, e.g., speed up, work, load balancing. A typical result is: *If $g(P)$ is increasing and convex, then the work per thread, $W(P)$, is convex. Let P^* be the unique solution of $P = [W(1)/\theta + g(P)]/g'(P)$, then the speed up is a maximum at P^*.* Here θ is the additional work required of a thread when an event occurs and is characteristic of the hardware and operating system.

The iteration considered is the *Jacobi SI* method applied to the linear system that arises from solving an elliptic PDE in two dimensions. This iteration is inherently parallel and can be summarized on a P processor machine as follows:

```
Initialize

For NITER = 1 to LAST

    For NPROC = 1 to P
        Do all iterations on my equations
        Send my boundary varizbles to neighbors
        Receive neighbors boundary variables
    End

End
```

The actual computation is complicated by steps to test for convergence and to adapt iteration parameters. These steps require global communication.

The NCUBE 1 used has 128 processors with both communication and computation handled by the single processor at each node. Communication is expensive on this machine relative to computation.

2. EXPERIMENTS AND MONITORING

The TRIPLEX tool set [1] is used to collect the trace data for two computations:

solid: PDE problem with 33 x 33 grid (about 1000 unknowns) on a nonrectangular domain (see Figure 5). Two to 128 processors used.

dash PDE problem with 50 x 50 grid (about 2500 unknowns) on a square. Four to 64 processors used.

The names solid and dash refer to the lines used in the graphical data given below. The communication procedure used is to send *all* data from one processor to all other processors (broadcast). This inherently inefficient scheme was used because (a) the mapping of subdomains to processors was not known, (b) the system utilities for multicasting do not work properly, (c) we wanted to guarantee that the converges properties of the iterative method remain the same. Time is measured in units of a *tick* which is about 0 1 msec. Five iterations were made.

3. MEASUREMENTS AND DISCUSSION

Figure 1 shows the most basic data obtained in the E/T methodology, the characteristic function $E = g(P)$, the number of events per thread (or processor). The average (expected) number of events per thread are shown as a function of P using solid and dash lines for the two experiments. The 95% level confidence intervals as computed from the experimental data are shown by + (for the solid case) and x (for the dash case). We see that $g(P)$ grows linearly with slope about two on this semi-logarithmic plot so $g(P) \sim O(P^2)$. It is known [2] that such a computation cannot exploit high parallelism well.

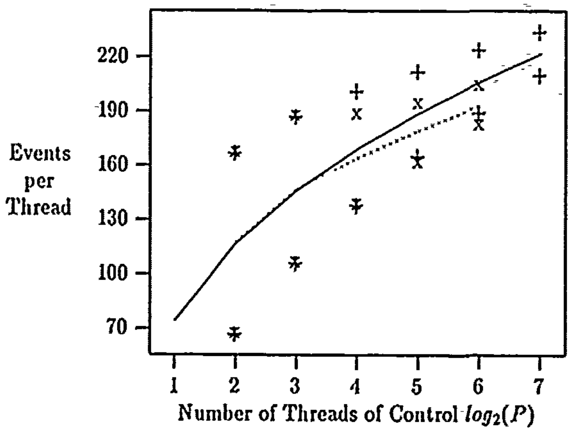


Figure 1: The expected number of *events* per thread of control and a 95% confidence interval for it.

Blocking is an important source of low performance, we illustrate the phenomena in Figure 2. At time t_1 processor P_i executes a READ and at time t_4, the requested result is available. The time $t_2 - t_1$ is *algorithmic blocking*, the time spent waiting because processor P_j has not yet computed the data P_i requests. The time $t_3 - t_2$ is *propagation delay*, the time spend waiting for the WRITE operation to be executed and the time $t_4 - t_3$ is *transmission delay*. The propagation and transmission delays on the NCUBE 1 are much larger than the ones in second and third generation hypercubes, but it is usually algorithmic blocking that causes very poor performance. The total time $t_4 - t_1$ is often called *synchronization delay*.

*This research supported in part by NSF grant CCR 86-19817 and by the Strategic Defense Initiative through ARO grants DAAG03-86-K-0106 and DAAL03-86-0107.

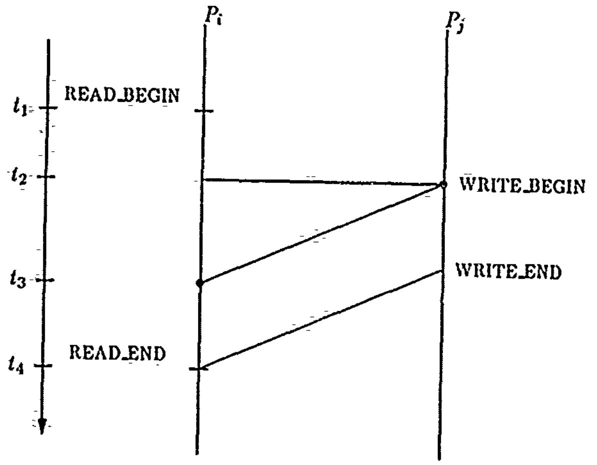


Fig:re 2: Communication involving blocking.

Figure 3 shows the expected algorithmic blocking time as a fraction of the total blocking time for these two PDE computations. The most obvious fact is that algorithmic blocking increases strongly as the number of processors increases. Note that the irregular behavior of these two curves are similar and can be explained from a detailed analysis of the characteristics of this computation [3].

Figure 4 shows the *computing time* as a fraction of the total non-blocked time per thread. Here time is partitioned into blocked (see Figure 2), performing I/O (Read or Write), and computing (presumably the useful work). We see that the fraction of computing time decreases rapidly with increasing P, in an ideal case one hopes for this to decrease much more slowly.

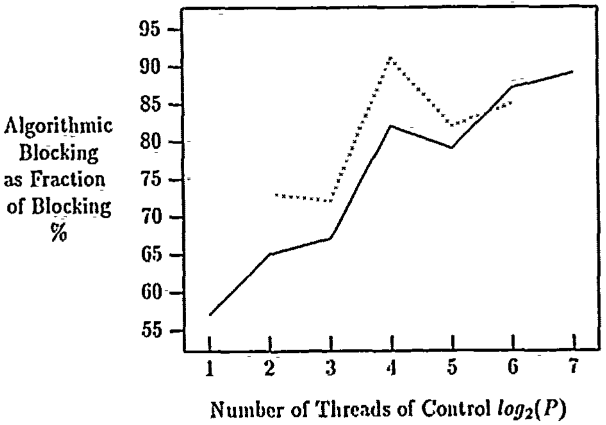


Figure 3: The expected *algorithmic blocking time* during a read operation as a fraction of the total blocking time during a read operation.

4. CONCLUSIONS

Our E/T model predicts that parallelism cannot be effectively exploited if the characteristic function $g(P)$ is quadratic in P. Our experiments show that $g(P)$ is quadratic and that the speed up obtained is low. Further analysis of this data [4] provi les other conclusions, namely. (1) The cost of synchronization of the iteration is quite high, (2) The Jacobi SI code adapts the iteration parameters and tests for termination often, this results in even poorer performance due to the relative inefficiency of these computations f r a distributed memory machine. A self-synchronization approach is presented and discussed in [2] which can alleviate these synchronization delay problems. Other approaches to improve performance are presented in [3].

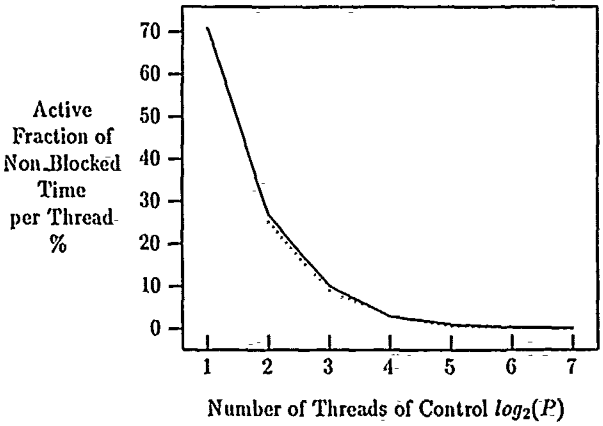


Figure 4: The expected *computing time* as fraction of the non-blocked time per thread.

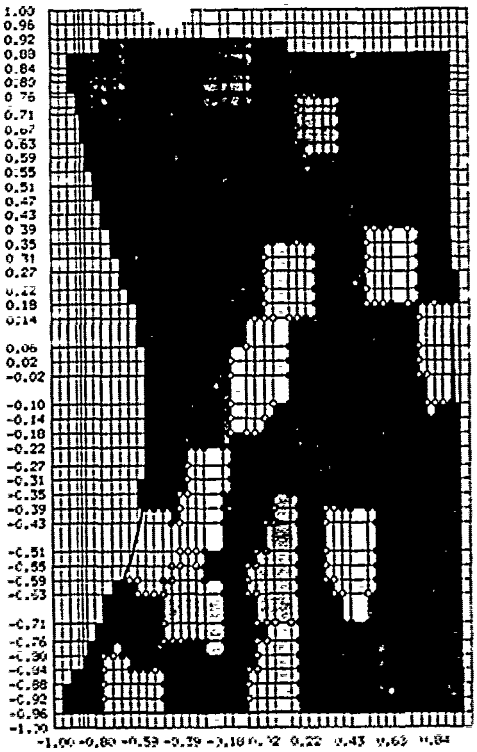


Figure 5: The nonrectangular PDE domain and its decomposition.

5. REFERENCES

[1] D. Krumme, A. Couch, B. House, and J. Cox. The Triplex Tool Set for the NCUBE Multiprocessor, Technical Report, Department of Computer Science, Tufts University, Medford, MA, 1989.

[2] D.C. Marinescu and J.R. Rice, On High Level Characterization of Parallelism, CSD-TR-1011, Computer Science Dept., Purdue University, August, 1990, 25 pages.

[3] D.C. Marinescu, J.R. Rice and E.A. Vavalis, Performance of Iterative Methods for Distributed Memory Machines, CSD-TR-979, Computer Science Dept., Purdue University, Septembe., 1990, 24 pages.

[4] D.C. Marinescu and J.R. Rice, The Effects of Communication Latency Upon Synchronization and Dynamic Load Balance on a Hypercube, *Proc. Intl. P..r. Proc. Symp.*, IEEE Press, 1991, to appear.

PARALLELIZING ITPACKV 2D

DAVID R. KINCAID
Center for Numerical Analysis
University of Texas at Austin
Austin, Texas 78713-8150 USA

and

MALATHI RAMDAS
National Instruments
6504 Bridge Point Pkwy
Austin, Texas 78731 USA

Abstract. ITPACKV 2D is a research-oriented numerical software package of iterative algorithms for solving large sparse systems of linear equations developed in the Center for Numerical Analysis, The University of Texas at Austin. The intent of this paper is to report on the project of parallelizing this software on the eight-processor Cray Y-MP supercomputer using its advanced multitasking facilities. Model problems in two-dimensions and three-dimensions are used to test the performance of the parallelized version of the package using one to eight processors. Numerical results are used for comparing the performance of the routines in this package, for determining the speedup ratios for them, and for drawing general conclusions with regard to the efficiency and parallelizability of each of the methods.

I. Introduction. The ITPACK Project was started over a decade ago in the Center for Numerical Analysis of The University of Texas at Austin to conduct basic research on iterative algorithms for solving large sparse systems of linear algebraic equations The emphasis has been on developing, testing, and evaluating software for solving linear systems arising from partial differential equations discretized using finite-differences and/or finite-elements. Several ITPACK packages have been developed, modified, improved, and changed through various versions.

In the ITPACKV 2D package, the basic iterative methods [Jacobi (J) Reduced System (RS), and Symmetric Successive Overrelaxation (SSOR)] are combined with two acceleration procedures; namely, Chebyshev (SI) and Conjugate Gradient (CG). Also, included in the package is the Successive Overrelaxation (SOR) method. The package uses adaptive procedures for the selection of the acceleration parameters and for automatic stopping tests. These routines work best when solving systems with symmetric positive definite or mildly nonsymmetric coefficient matrices.

The basic iterative routines available in the package are as follows.

JCG	Jacobi Conjugate Gradient
JSI	Jacobi Semi-Iteration
SOR	Successive Overrelaxation
SSORCG	Symmetric SOR Conjugate Gradient
SSORSI	Symmetric SOR Semi-Iteration
RSCG	Reduced System Conjugate Gradient
RSSI	Reduced System Semi-Iteration

Two orderings of the unknowns in the linear system are available: the natural (lexicographic) ordering and the red-black (checker-board) ordering.

II. Vectorization. ITPACKV 2D [8, 10] is a modified version of ITPACK 2C [11] with enhanced vectorization capabilities. The primary changes made for the vectorized version [13] are

1. changing the storage format for the coefficient matrix from the "Yale sparse storage format" [4] to the "ELLPACK matrix storage format" [15] since the latter is more vectorizable, and
2. using a *wavefront ordering* (ordering by diagonals) to enable SOR, SSORCG, and SSORSI routines to vectorize under natural ordering.

The primary vectorizable and parallelizable operations in the iterative algorithms in ITPACKV 2D [7] are matrix-vector multiplications, forward solves and backward solves.

A. Matrix-Vector Multiplication. The Yale sparse matrix format [4], as used in ITPACK 2C, is a row-wise storage format using three linear arrays a, ja, ia. With this data structure, a matrix vector multiplication of the form Au results in operations where the maximum vector length is equal to the number of nonzero elements in the row, which may be small for sparse matrices.

In the ELLPACK sparse matrix storage format [15], a rectangular array coef stores the nonzero elements of the scaled coefficient matrix A in a row-wise fashion. [The original linear system $Ax = b$ is initially scaled by the diagonal $D = \text{diag}(A)$ with $(D^{-1/2}AD^{-1/2})(D^{1/2}x) = (D^{-1/2}b)$ so that it assumes the form $(I - G)u = k$. The entries in coef are also re-ordered when the red-black or wavefront ordering is used.] Another rectangular array jcoef stores the column numbers of the corresponding elements in coef. If maxnz is the maximum number of nonzero elements per row in A and n is the number of columns, then the matrix-vector product $w \leftarrow k+Gu$ can be computed with this data structure as follows.

```
      do 20 j = 1,maxnz
         do 10 i=1,n
            w(i) = rhs(i)
10       continue
         do 15 i = 1,n
            w(i) = w(i) + coef(i,j)*u(jcoef(i,j))
15       continue
20    continue
```

This approach has the advantage that the vector lengths are long (equal to n, the order of the system) and only maxnz gather operations are required. Hence, there are fewer vector start-up costs compared to the original format used in this package.

B. Forward and Backward Solves. Forward/Backward solutions of sparse triangular matrices are a major component of the SOR and SSOR routines with the natural ordering. Since these routines do not vectorize in this situation, the unknowns are re-organized using a wavefront ordering. Several authors [1, 2, 3, 16] have noted that this ordering can effectively vectorize the forward solution process of a 5-point finite-difference stencil on a rectangular grid.

Although this concept of wavefront ordering is applied in ITPACKV 2D to vectorize the SOR and SSOR routines in the case of natural ordering, the vector lengths are still much shorter than for the Jacobi or the Reduced System routines.

III. Microtasking. The basic parallelizing technique used is to partition do-loops into segments. This partitioning is done by introducing an outer loop that defines these partitions and assigns them to available processors. The beginning of a parallel do-loop is marked using a control command directive.

For example, the computation of a matrix-vector product $w \leftarrow k + Gu$ is shown below.

```
CHIC$ DO ALL SHARED (ntask,n,maxnz,u,w,rhs,coef,jcoef),
CHIC$*       PRIVATE (k,i,j,ist,ied)
         do 25 k = 1,ntask
            ist = ((k-1)*n)/ntask + 1
            ied = (k*n)/ntask
            do 10 i = ist,ied
               w(i) = rhs(i)
10          continue
            do 20 j = 1,maxnz
               do 15 i = ist,ied
                  w(i) = w(i) + coef(i,j)*u(jcoef(i,j))
15             continue
20          continue
25       continue
```

Here ntask is the number of available processors. The CHIC$ DO ALL directive enables parallel execution using several processors with each computing a range of rows determined by the variables ist and ied. For this directive, the scope of each variable (shared and private) used within the region must be defined. Notice that no synchronization is needed within the loops. Forward and backward solves are also parallelized using the same basic idea of partitioning the outer loop.

For the iterative algorithms under red-black ordering, the matrix-vector multiplication can be partitioned in such a way that all the red points are updated in parallel and then all the black points. Thus, a single synchronization point is needed. In the Reduced System method based on the black points, for example, the computation is $w_R \leftarrow c_R+F_R u_B$, $w_B \leftarrow c_B+F_B w_R$ and it can be carried out as shown in the displayed code below.

```
CHIC$ PARALLEL SHARED (ntask,nr,nb,maxnz,w,rhs,u,coef,jcoef),
CHIC$*         PRIVATE (k,i,j,ist,ied)
CHIC$ DO PARALLEL
         do 25 k = 1,ntask
            ist = ((k-1)*nb)/ntask + nr + 1
            ied = (k*nb)/ntask + nr
            do 10 i = ist,ied
               w(i) = rhs(i)
10          continue
            do 20 j = 2,maxnz
               do 15 i = ist,ied
                  w(i) = w(i) + coef(i,j)*u(jcoef(i,j))
15             continue
20          continue
25       continue
CHIC$ END DO
```

```
CHIC$ DO PARALLEL
        do 45 k = 1,ntask
           ist = ((k-1)*nr)/ntask + 1
           ied = (k*nr)/ntask
           do 30 i = ist,ied
              v(i) = rhs(i)
30         continue
           do 40 j = 2,maxnz
              do 35 i = ist,ied
                 v(i) = v(i) + coef(i,j)*v(jcoef(i,j))
35            continue
40         continue
45      continue
CHIC$ END DO
CHIC$ END PARALLEL
```

IV. Model Problems. The parallelized version of ITPACKV 2D was tested on the Cray Y-MP and was applied to sparse matrix problems resulting from a finite-difference discretization of partial differential equations in two-dimensions (2-D) and three-dimensions (3-D). The two model problems are described in this section.

2-D Model Problem

$$\begin{cases} u_{xx} + 2u_{yy} = 0, & (x,y) \in \Omega_2 = (0,1) \times (0,1), \\ u = 1 + xy, & (x,y) \in \partial\Omega_2. \end{cases}$$

This problem was discretized using the standard 5 point finite-difference stencil with mesh sizes $h = 1/100$ and $h = 1/140$ resulting in systems of size $N = 99^2 = 9,801$ and $N = 139^2 = 19,321$, respectively The stopping criterion used is given by $[1/(1 - M_E)] \| \delta^{(n)} \|_2 / \| u^{(n)} \|_2 \leq \zeta$, where $\delta^{(n)} = Gx^{(n)} + k - x^{(n)}$ is the *pseudo-residual* at the nth iteration, $u^{(n)}$ is the nth iterative approximate solution vector, and M_E is an estimate of the maximum eigenvalue of the iteration matrix G [5]. For this stopping test, ζ was taken to be 10^{-6}.

3-D Model Problem

$$\begin{cases} u_{xx} + 2u_{yy} + u_{zz} = 0, & (x,y,z) \in \Omega_3 = (0,1) \times (0,1) \times (0,1), \\ u = 1, & \text{if } x = 0 \text{ or } y = 0 \text{ or } z = 0, \\ u_x = yz(1 + yz), & \text{if } x = 1, \\ u_y = xz(1 + xz), & \text{if } y = 1, \\ u_z = xy(1 + xy), & \text{if } z = 1. \end{cases}$$

When the standard 7-point finite-difference stencil was used with mesh sizes $h = 1/20$ and $h = 1/30$, the resulting problem sizes were $N = 20^3 = 8,000$ and $N = 30^3 = 27,000$, respectively.

For both problems, all the routines were tested using both natural ordering and red-black ordering of the unknowns. The various iteration parameters needed in the algorithms were determined using adaptive procedures. The only exceptions were the SSORCG and SSORSI algorithms where the relaxation factor ω was fixed to be 1 for the red-black ordering.

V. Results and Discussion. Among the different multitasking tools provided by the Cray computer systems, *microtasking* seems best suited for parallelizing ITPACKV 2D since this software has the potential for parallelism more at the do-loop level. The basic principles followed in *microtasking* this package involved partitioning the loops into equal segments that are assigned to available processors and combining do-loops into long parallel regions, wherever possible, in order to reduce parallel startup costs

The model problems in 2-D and 3-D given above were used to test the performance of the *microtasked* version for varying number of processors. The maximum number of processors used was eight. All the tests were run in dedicated mode. The numerical results are presented in [14]

In comparing the timings between the uni-processor version and the sequential version, it was observed that the sequential code was faster than the uni-processor version of the parallelized code for most all of the routines This is due to the overhead costs associated with the loop partitioning in the parallelized code.

With regard to overall speed, SSORCG and SSORSI were the fastest routines using only a single processor for the smaller 2-D problem However for eight processors, JCG was the fastest due to its superior parallelization. For the larger 2-D problem, SSORSI was the fastest on a single processor and JCG performed the best for eight processors. The fastest routines using one processor for the smaller 3-D problem were JCG and SSORCG under natural ordering. For the larger problem, SSORCG was the fastest routine. For eight processors, JCG gave the best result for both problems since it parallelized very well. Under red-black ordering, all the routines performed reasonably well with the RSCG being the fastest routine for all the problems for any number of processors.

The speedup ratios obtained for the larger problems (both 2-D and 3-D) were better than that obtained for the smaller problem. Under natural ordering, the JCG and JSI routines gave good speedup results while the SOR and SSOR routines performed quite poorly. In general, the wavefront ordering used for these routines resulted in short vector lengths and also required the processors to synchronize too often. However, the SOR and SSOR routines had improved speedups under the natural ordering for the 3-D problems. This was due to the fact that there were fewer wavefronts for the 3-D problems in comparison to the 2-D problems. This minimized the number of synchronization points. Under red-black ordering, all routines had good speedups. For both ordering, the speedup ratios obtained for the 3-D problems were not as good as those obtained for the 2-D problems.

Since the largest speedup does not imply the fastest procedure, the recommended routines in this parallel package are RSCG when the unknowns can be re-ordered into the red-black ordering and JCG otherwise.

Acknowledgements. The authors gratefully acknowledge the helpful advice and useful suggestions given to them by Tom Oppe, Irina Mukherjee, and David Young.

This work was accomplished with the aid of the Cray Y-MP supercomputer at Cray Research, Inc., Mendota Heights, MN, and with the Cray Y-MP computer operated by the Center for High Performance Computing of The University of Texas System. The cooperation of each of these centers in making their facilities available is greatly appreciated.

The authors wish to acknowledge the Center for Numerical Analysis and the Computation Center of The University of Texas at Austin for their support. In addition, this work was supported, in part, by the National Science Foundation under grant CCR-8518722, the Department of Energy under grant DE-FG05-87ER25048, and Cray Research, Inc., under grant LTR DTD with The University of Texas at Austin.

REFERENCES

[1] Adams, L., "Reordering Computations for Parallel Computation," *Comm. Appl. Numer. Methods* 2, 263–271, 1986.

[2] Ashcraft, C., "Moving Computation Front Approach for Vectorizing ICCG Calculations," General Motors Publication, GMR-5174, 1985.

[3] Ashcraft, C., Shook, M., Jones, J., "A Computational Survey of Conjugate Gradient Preconditioners on the Cray 1-S," General Motors Research Publication, GMR-5299, 1986.

[4] Eisenstat, S. C., Gursky, M. C., Schultz, M. H., Sherman, A. H., "Yale Sparse Matrix Package: The Symmetric Codes," Research Report #112, Department of Computer Science, Yale University, May 1977.

[5] Hageman, L. A., Young, D. M., *Applied Iterative Methods*, Academic Press, New York, 1981.

[6] Kincaid, D. R., Oppe, T. C., "The Performance of ITPACK on Vector Computers for Solving Large Sparse Linear Systems Arising in Sample Oil-Reservoir Simulation Problems," *Comm. Appl. Numer. Methods* 3, 23–29, 1987.

[7] Kincaid, D. R., Oppe, T. C., "Recent Vectorization and Parallelization of ITPACKV," Report CNA-233, Center for Numerical Analysis, University of Texas at Austin, November 1989. (Also in *Preconditioned Conjugate Gradient Methods*, O. Axelsson and L. Yu. Kolotilina (eds.), Lecture Notes in Mathematics 1457, Springer, New York, 58–78, 1990.)

[8] Kincaid, D. R., Oppe, T. C., Young, D. M., "ITPACKV 2C User's Guide," Report CNA-191, Center for Numerical Analysis, University of Texas at Austin, February 1984.

[9] Kincaid, D. R., Oppe, T C , Young, D. M., "Vector Computations for Sparse Linear Systems," *SIAM J. Alg. Disc. Math.* 7, 99–112, 1986.

[10] Kincaid, D. R., Oppe, T. C., Young, D. M., "ITPACKV 2D User's Guide," Report CNA-232, Center for Numerical Analysis, University of Texas at Austin, May 1989.

[11] Kincaid, D. R., Respess, J. R., Young, D. M., Grimes, R. G., "ITPACK 2C: A FORTRAN Package for Solving Large Sparse Linear Systems by Adaptive Accelerated Iterative Methods," *ACM Trans. Math. Software* 8, 302–322, September 1982.

[12] Kincaid, D. R., Young, D. M., "A Brief Review of the ITPACK Project," *J. Comp. and Appl. Math.* 24, 33–54, 1988.

[13] Oppe, T. C., "The Iterative Solution of Large Sparse Linear Systems Using Vector Computers," Report CNA-241, Center for Numerical Analysis, University of Texas at Austin, February 1990.

[14] Ramdas, M., "Parallelizing ITPACKV 2D for the Cray Y-MP," M. A. Report, University of Texas at Austin, December 1990. (Also, Report CNA-249, Center for Numerical Analysis, University of Texas at Austin, February 1991.)

[15] Rice, J. R., Boisvert, R. F., "Solving Elliptic Problems using ELLPACK," Springer-Verlag, New York, 1985.

[16] Van der Vorst, H. A., "(M)ICCG for 2D Problems on Vectorcomputers," Report No. A-17, Data Processing Center, Kyoto University, Kyoto, Japan, 1986.

[17] Young, D. M., *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

Rectilinear Partitioning of Irregular Data Parallel Computations

David M. Nicol*
Department of Computer Science
College of William and Mary
Williamsburg, VA. 23185 USA
Internet: nicol@cs.wm.edu

Introduction

This paper describes new mapping algorithms for domain-oriented data-parallel computations, where the workload is distributed irregularly throughout the domain. We consider the problem of partitioning the domain (represented as an $n \times m$ array of execution weights) for an $N \times M$ array of locally connected parallel processors, in such a way that the workload on the most heavily loaded processor is minimized, subject to the constraint that the partition be perfectly rectilinear, as illustrated by Figure 1. The total execution weight of a processor is the sum of all execution weights assigned to its rectangle. Data parallel computations tend to have localized data dependencies, rectilinear partitions ensure that all communication induced by localized data dependencies is between local processors, this algorithm will be particularly useful on architectures where there is a high differential between the cost of local and global communication. We will not explicitly include communication costs in our model. Our implicit assumption is that communication costs will be minimized when locality is ensured.

This paper provides an improved algorithm for finding the optimal partition in one dimension, new algorithms for partitioning in two dimensions, and shows that optimal partitioning in three dimensions is NP-complete. We discuss our application of these algorithms to real problems.

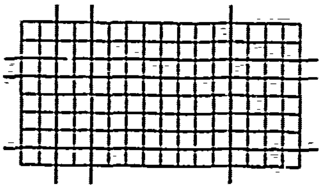


Figure 1. Rectilinear Partitioning of Two Dimensional Domain

One Dimensional Partitioning

The rectilinear partitioning algorithm in one dimension has been extensively studied as the *chains-on-chains* partitioning problem [1, 3, 4, 5]: we are given a linear sequence of work pieces (called modules), and wish to partition the sequence for execution on a linear array of processors. The best known published algorithm to date finds the optimal partitioning in $O(Mm \log m)$ time, where M is the number of processors and m is the number of modules. This solution repeatedly calls a probe function. This function accepts an argument W, and uses a greedy workload assignment algorithm to determine whether there is a partition that assigns no more than W work to each processor. probe is used in conjuction with a search, in order to find the minimal W for which there exists a feasible partition. The cost of calling probe is $O(M \log m)$. Previous solutions have involved calling probe $O(m)$ times. We have a new searching strategy that reduces the calling frequency to $O(M \log m)$, thereby reducing the complexity of one-dimensional partitioning to $O(m + M^2 \log^2 m)$.

Two dimensional Partitioning

The heart of our 2D partitioning algorithms is an ability to optimally partition in one dimension, given a fixed partition in the other. Suppose a row partition R is given. We can compress workload forced (by R) to reside on a common processor into super-pieces, thereby creating an $N \times m$ load matrix. This matrix can be viewed as N one dimensional chains, a common partitioning of their columns will produce a 2D rectilinear partition.

The problem of finding an optimal column partition can be approached through a minor modification to the 1D probe function. This modification raises the cost of calling probe to $O(NM \log m)$, plus an $O(nm)$ preprocessing cost. Otherwise, the conditionally optimal problem is solved in the same way as the the 1D partitioning problem, in $O(mn + NM^2 \log^2 m)$ time.

We may apply the conditionally optimal partitioning algorithm in an iterative fashion. Suppose that a row partition R_1 is given. For example, we might construct an initial row partition as follows. sum the weights of all work pieces in a common row, to create a super-piece representing that row. Find an optimal 1D partition of those super-pieces onto N processors. Use this partition as R_1, assume it to be fixed, and let C_1 be the optimal column partition, given R_1. Let $\pi_1 = \pi(R_1, C_1)$ be the cost of that partitioning. Next, fix the column partition as C_1, and let R_2 be the optimal row partitioning, given C_1. Let $\pi_2 = \pi(R_2, C_1)$. Clearly we may repeat this process as many times as we like. We have shown that the sequence $\pi_1, \pi_2, \ldots$, is monotone non-increasing, and that eventually the computation converges to a fixed row partition R_∞ and column partition C_∞. We have also bounded the number of iterations required for convergence by $O(n^2m^2(n+m))$. Far fewer iterations are required to converge, in practice. The talk will discuss the use of this procedure on highly irregular 2D grids used in fluid-flow problems.

Three Dimensional Partitioning

Finally, we consider the complexity of the 3D partitioning problem. We have already seen that the 1D problem can be solved in polynomial time; it is not yet known whether the 2D problem is tractable. It turns out that the problem of finding an optimal rectilinear partition in three dimensions is NP-complete. The proof shows that the monotone 3SAT problem [2] can be reduced to rectilinear partitioning in three dimensions. The key idea is to construct a domain as a function of the 3SAT clauses. Each literal is given three rows, or columns in the 3D weight matrix. The intersection of rows and columns for literals x_i, x_j, x_k is a $3 \times 3 \times 3$ volume. This volume is to be partitioned among $2 \times 2 \times 2$ processors, which essentially forces each literal group to divide into one of two possible partitionings. The partitioning choice can be interpreted as the assignment of a truth value to the literal. The volume is weighted in such a way that its partition has bottleneck value 1 if and only if the partition corresponds to an assignment that satisfies the clause. This shows that optimal three dimensional partitioning is as hard as the monotone 3SAT problem, which is known to be NP-complete.

References

[1] S. H. Bokhari. Partitioning problems in parallel, pipelined, and distributed computing. *IEEE Trans. on Computers*, 37(1):48-57, January 1988.

[2] M.R. Garey and D.S. Johnson. *Computers and Intractability*. W.H. Freeman and Co., New York, 1979.

[3] M.A. Iqbal. Approximate algorithms for partitioning and assignment problems. Technical Report 86-40, ICASE, June 1986.

[4] M.A. Iqbal and S.H. Bokhari. Efficient algorithms for a class of partitioning problems. Technical Report 90-49, ICASE, July 1990.

[5] D.M. Nicol and D.R. O'Hallaron. Improved algorithms for mapping parallel and pipelined computations. *IEEE Trans. on Computers*, 1991.

*This research was supported in part by the Army Avionics Research and Development Activity through NASA grant NAG-1-787, in part by NASA grant NAG-1-1132, and in part by NSF Grant ASC 8819373.

Applying Chain Mapping Algorithms to Flowgraphs*

David R. O'Hallaron

School of Computer Science
Carnegie Mellon University, Pittsburgh, PA, 15213, USA

Abstract

We identify two useful classes of flowgraphs that can be compiled onto distributed–memory multicomputers using chain mapping algorithms, and whose performance can be predicted accurately.

1 Introduction

Flowgraphs have been used for years to model digital signal processing (DSP) algorithms. A *flowgraph* is a collection of nodes and arcs, where nodes represent computations and arcs represent FIFO queues. Each node iterates an infinite number of times; each iteration, a node consumes a vector of data items from each of its input arcs, performs a computation on the input vectors, and produces a vector of data items on each of its output arcs. The sizes of the input and output vectors are indicated by integer arc labels.

There is much published research on the problems of compiling flowgraphs onto sequential machines and onto shared–memory multiprocessors[3]. Little attention has been given to date on the problem of compiling flowgraphs onto distributed memory multicom puters; some recent work can be found in [5, 8]

In this paper we identify two useful classes of flowgraphs that we call *pipes* and *trellises*. A pipe models a pipelined sequence of different operations performed on a single data stream. A trellis models a single operation performed in parallel on different data streams. Pipes and trellises are interesting because they can be compiled efficiently and optimally using chain mapping algorithms, and because their performance can be accurately predicted.

2 Pipes and Trellises

A common DSP application is to pipeline data from a sensor through a sequence of filtering operations such as FIR filters and FFT's. Such applications are modeled by a flowgraph we call a *pipe*. An example with 4 filtering operations is shown in Figure 1.

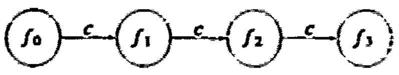


Figure 1: A pipe.

In another common DSP application, data from multiple sensors is distributed, processed independently and combined in some way. Such applications can be modeled by a flowgraph we call a *trellis*. An example with four sensors is shown in Figure 2.

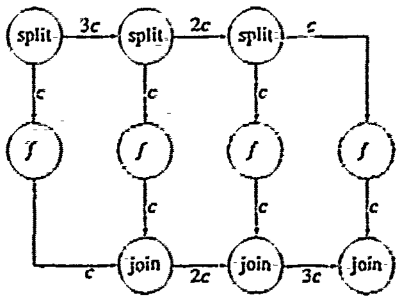


Figure 2: A trellis.

Pipes and trellises are extremely useful classes of flowgraphs that can be used to build real applications. For example, sonar adaptive beam interpolation and the 2D FFT can each be modeled as a pipe of two trellises.

3 Applying Chain Methods

Chain mapping algorithms[1, 2, 4] compile a chain of M modules onto a chain of P distributed–memory processors, typically with $P \leq M$. These algorithms are attractive because they are efficient, requiring time polynomial in M and P, and because they are optimal, minimizing the maximum load on any processor, subject to the constraint that two contiguous modules are mapped onto the same processor or its neighbor.

Pipes and trellises have the nice property that they are easily transformed into a form suitable for chain mapping methods. The transformation of pipes is trivial; each node in the flowgraph becomes a module for the chain mapping algorithm. This is shown in Figure 3. The transformation for trellises is also straightforward, as shown in Figure 4. Each of the dashed boxes becomes a module for the chain mapping algorithm.

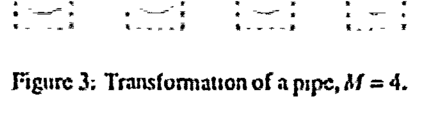


Figure 3: Transformation of a pipe, $M = 4$.

*The research was supported in part by Defense Advanced Research Projects Agency (DOD) monitored by DARPA/CMO under Contract MDA972-90-C-0035 and in part by the Office of Naval Research under Contract N00014-90-J-1939.

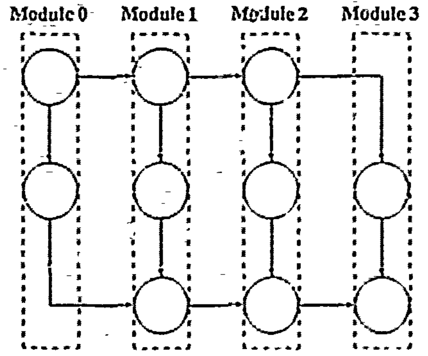


Figure 4: Transformation of a trellis, $M = 4$.

4 Predicting Performance

We have seen that pipes and trellis can be compiled onto distributed-memory multicomputers using chain mapping methods. Another nice quality of these flowgraphs is that their performance can be predicted quite accurately. A common performance measure for flowgraphs is *speedup*, denoted by S and defined by $S = E/P$, where E denotes *efficiency*. We derive an expression for the efficiency of pipes and trellises under the following assumptions: (1) All modules consume the same number of clocks per iteration. (2) The number of modules M is an integral multiple of the number of processors P.

Let $T = T_c + T_d$ be the number of number of clocks per iteration per module, where T_c is the number of clocks spent doing computations and T_d is any additional clocks required because of interprocessor communication, these can be clocks spent waiting for data or actually transferring data between a communications network and memory. Under these assumptions, efficiency is simply

$$E = \frac{T_c}{T_c + T_d} = \frac{1}{1 + T_d/T_c} \tag{1}$$

If the computations associated with each node in the flow graph are data–independent, true for most DSP operations, then T_c can be predicted precisely for a given parallel computer and compiler.

On the other hand, T_d may be more difficult to predict accurately because it could include time spent waiting for data to arrive. For our model we will assume that $T_d = 2\tau N$, where N is the number of inputs (and outputs) for each module, and τ is the overhead (in clocks) associated with transferring a word between the communication network and memory. Notice that τ is constant for a given parallel machine and compiler. Notice also that this expression for T_d is a lower bound that ignores any clocks spent waiting for data to arrive. While this is unrealistic in general, empirical evidence suggests that this is reasonable for pipes and trellises. Given the expression for T_d, efficiency becomes

$$E = \frac{1}{1 + 2\tau N/T_c} \tag{2}$$

We tested the model in (2) using four different pipes and trellises running on a 64–processor iWarp computer at Carnegie Mellon[6, 7]. Each flowgraph was compiled using a chain mapping algorithm, and the resulting chain was embedded in the iWarp's 2D mesh. The results are shown in Figure 5. Predicted efficiency was computed using (2). For all tests, $P = 64$, $M = 64$, and $\tau = 40$. Measured efficiency was obtained by measuring the speedup S and then applying the identity $E = S/P$.

Graph	N	T_c	Predicted E	Measured E
pipe1	1024	$55N$	41%	40%
pipe2	1024	$23N^2$	99%	99%
trellis1	2048	$42N$	34%	34%
trellis2	2048	$41N + 562(N/M)^2$	80%	80%

Figure 5: Predicted and measured efficiency on iWarp

5 Discussion

The empirical results are somewhat startling. In each case, the predicted performance was within one percent of the measured performance. Communications overhead was restricted to the unavoidable cost of physically transferring data between the communications network and memory. There was negligible overhead due to waiting for data.

The accuracy of the model allows us to make strong predictions about performance for various values of τ, N, and T_c. For example, if τ could be reduced from 40 to 4, which is quite possible for iWarp, then the efficiency of the *trellis1* flowgraph in Figure 5 would rise from 34% to 84%.

References

[1] S. H. Bokhari. Partitioning problems in parallel, pipelined and distributed computing. *IEEE Transactions on Computers*, 37(1):48–57, January 1988.

[2] M. Iqbal. Approximate algorithms for partitioning and assignment problems. Technical Report 86-40, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, June 1986.

[3] E. A. Lee and D. G. Messerschmitt. Static scheduling of synchronous data flow programs for digital signal processing. *IEEE Transactions on Computers*, C-36(1):24–35, January 1987.

[4] D. M. Nicol and D. R. O'Hallaron. Efficient algorithms for mapping pipelined and parallel computations, 1991. to appear in IEEE Transactions on Computers.

[5] H. W. Printz. *Automatic Mapping of Large Signal Processing Systems to a Parallel Machine.* PhD thesis, Carnegie Mellon, January 1991.

[6] S. Borkar et. al. iWarp: An integrated solution to high-speed parallel computing. In *Supercomputing '88*, Kissimmee, FL, November 1988.

[7] S. Borkar et. al. Supporting systolic and memory communication in iWarp. In *17th Annual International Symposium on Computer Architecture*. IEEE Computer Society and ACM, May 1990.

[8] G. C. Sih and E. A. Lee. Scheduling to account for interprocessor communication within interconnection-constrained processor networks. In *Proceedings of the 1988 International Conference on Parallel Processing*, August 1990.

A MAPPING ALGORITHM FOR HETEROGENEOUS MULTIPROCESSOR ARCHITECTURES

Todd P. Carpenter
Honeywell Systems and Research Center
3660 Technology Drive
Mpls. MN 55418-1006 USA
carpent@src.honeywell.com

Sudhakar Yalamanchili
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA. 30332-0250, USA
sudha@eecom.gatech.edu

Abstract-Resource management is an integral facet of the design, development, and application of multiprocessor architectures. This paper is concerned with one specific aspect of the problem - the assignment of concurrently executable (within precedence constraints) tasks to the processors. Naive assignments can lead to excessive performance degradation due to the consequent overhead in inter-task communication and synchronization. In particular we focus on heterogeneous architectures comprised of distinct processor types, distinct communication media, and many data types. Such systems are becoming increasingly widespread, particularly in special purpose applications and distributed systems. We address the problem of statically mapping programs to such multiprocessors.

I. INTRODUCTION

Examples of heterogeneous applications include distributed environments - different processor platforms, mix of workstations, mini-, minisuper-, mainframe computers, etc. They represent different processors of varying performance and compilers may or may not exist for all languages on all machines. Further, each application module runs at differing speeds on differing processors. Each pair of processors may communicate at different rates. For instance, heterogeneous environment might consist of Apollo, SUN, Decstations, Vaxen and Amiga platforms. Alternatively, dedicated applications often are supported by heterogeneous multiprocessor architectures to realize high performance. For example, The Intel IPSC/2 hypercube is available as mixed speed systems using i860 and 80386 processors. Packaging constraints often lead to systems where subcubes of a hypercube residing in different racks communicate over different media (e.g, optic fiber) than processors within a rack. Special purpose multiprocessor architectures for signal processing have been designed with hierarchically organized interconnection networks with differing cost/delay characteristics at each level.

The problem of mapping parallel programs onto homogeneous architectures is a sufficiently complex combinatorial problem in its own right [1]. The presence of additional constraints of processor and processing types and variable communication delays makes the problem even more difficult. In this paper we describe an approach for realizing such assignments. This approach is an adaptation of an approach we successfully implemented for computing assignments when the target architecture (and the computations) are homogeneous [2]. This work only applies to the static assignment problem. We do not yet address the more difficult dynamic mapping problem.

II. MODEL AND METHODS

The mapping problem is addressed in the context of exploiting medium-coarse grained parallelism in message passing. multiple instruction stream multiple data stream (MIMD) architectures. The application and and architecture are described in terms of attributed, directed graphs. Node (edge) weights represent computing (communication) requirements. Nodes are also labeled by a type representing the nature of the computation, e.g., floating point, symbolic, etc. Assignments must now maintain compatibility between types, i.e. a node of a certain type is constrained to execute on only a subset of processors. We propose to compute assignments based on an adaptation of simulated annealing - a combinatorial optimization procedure that realizes a probabilistic search of a discrete state space [3]. The process starts from some initial state which is perturbed. The new state is evaluated based on an objective function (also known as the energy function). If this state is "better" as measured by the value of the energy function it is accepted. If not it is probabilistically accepted. It is this latter random behavior that enables the search process to avoid being trapped in local minima. Simulated annealing has found wide application in a number of combinatorial optimization problems. A simplified view of the process is shown below, and each step is elaborated in the following.

```
Loop until done
  state perturbation;
  state evaluation;
  state acceptance/rejection;
end loop
```

A. State Perturbation

Given an assignment of tasks to processors, this step modifies the assignment(s) of task(s). The change may be based on architectural characteristics (e.g., limited to adjacent processors) or may be based on the structure of algorithm (e.g., limited to communicating tasks). In a single iteration, multiple tasks may be moved simultaneously corresponding to large jumps in the state space, or one task may be moved at a time. Further, the perturbation strategies may change over time to adapt to the behavior of the search. Finally, the perturbations are governed by type constraints i.e., between tasks and processors.

B. State Evaluation

The energy function maps states to real or integer values. The value of the function is a measure of the quality of the mapping represented by that state. For the mapping problems we are considering examples of appropriate energy functions include maximum processor load, maximum inter-processor communication load, variance of processor loads, etc. The more complex the energy function the longer the run-times, or the smaller number of states evaluated per unit time. The goal is to find a state that maximizes (or minimizes) the value of the energy function.

We propose an energy function that we have successfully used when mapping onto homogeneous architectures. This function consists of two components - a load balancing component and a inter-processor communication component. It is typically desirable to maximize the first and minimize the second, and can be represented as, $E = w_{lb} \times LB + w_c \times C$ where w_{lb}, w_c are the weighting factors for load balancing and inter-processor communications. The individual components are computed as follows.

Load balance is a measure of how evenly the processing load is distributed among the processors. The objective is to distribute the processing load as evenly as possible. The value of the expression *LoadBalance* shown below is minimized when the load is evenly distributed.

Define:

P = number of resource elements

T_i = number of tasks mapped to element i

$Tw_{j,i}$ = the weight of the jth task mapped to resource element i

$Pw_{i,type}$ = the weight for resource element to perform *type* task

$Tt_{j,i}$ = the type of task j mapped to element i

LR = The number of resource elements with nonzero loads

$$LoadBalance = \sqrt{\frac{\sum_{i=1}^{P}(\sum_{j=1}^{T_i}(Tw_{j,i} \times Pw_{i,Tt_{j,i}}))^2}{LR}}$$

The expression *Communication* shown below is a measure of how the inter-processor communications load is distributed. The load is a function of the distance between communicating tasks, the data volume, and link throughput. The value of *Communication* is minimized when all tasks are mapped to the same processor. Therefore this expression is at odds with the load balancing term described above.

Define:

T = number of tasks

To_i = number of outputs of task Ti

$L_{RTi}L_{RTj}$ = the links from resource to which Task i is mapped, to the resource for task j

$Tc_{i,j}$ = the amount of data from Task i to Task j

L_k reciprocal of throughput of link k

β Amount of communications bandwidth potential

$$Communications = \frac{\sum_{i=1}^{T}\sum_{j=1}^{To_i}\sum_{k=L_{RTi}}^{L_{RTj}} Tc_{j,i} \times L_k}{\beta}$$

C. State Acceptance/Rejection

New states are accepted or rejected based on the value of the energy function. We deviate from traditional annealing implementations with respect to this decision process. The following acceptance/rejection techniques were employed in our study. Note that although we can accept worse states, we do maintain a record of the minimum state found.

- Annealing If $\Delta E \leq 0$, the new state is accepted, and is used as the starting point for the next iteration. If $\Delta E > 0$, the new state is accepted probabilistically according to $P(\Delta E) = e^{-\frac{\Delta E}{kT}}$ where k is the Boltzmann's constant and T is the temperature.
- Kappa Sequence If $\Delta E \leq 0$, or if κ contiguous states have been rejected, accept the new state. The value of κ is increased if $\Delta E \leq 0$, and is decreased if κ states have been rejected. This function contains some notion of history, and modifies the behavior of the search based upon trends.
- Lambda Sequence If $\Delta E \leq 0$ accept the new state, or accept the new state based on a probability distribution function Λ. After κ states, Λ is modified based on computable trends in the sequence of states. This is an adaptive acceptance function which modifies its behavior based on some history of movement within the state space.

III. EXPERIMENTS

A major difficulty with mapping algorithms is in evaluating their performance. With no known general method of computing the optimal assignments for comparison purposes, we apply the mapping algorithm to problems with known analytical solutions. For example, we apply the mapping algorithm to the problem of finding assignments of hypercubes onto hypercubes, or meshes onto hypercubes. Such an approach runs the risk of of not accurately representing the performance of more general cases, however our implementation makes use of no information about the structure of the target architecture or algorithms. The expectation is that comparable performance is possible with relatively unstructured cases.

We have also experimented with heterogeneous systems, including bus based and hypercube based systems. The bus based system consisted of a serial backplane, and IO and processing nodes. The hypercube consisted of heterogeneous communication links and processors, where subcubes have one communication dimension faster than the other dimensions, and each subcube consists of a different type (throughput) of processor.

Applications mapped to these hardware resources included airplane flight managements control systems, sonar beamforming, and various benchmark examples such as weighted hypercube and binary tree graphs.

IV. RESULTS

The results of our experiments have been very encouraging. For relatively small systems ($<$ 32 processors) based on regular, symmetric network topologies such as the hypercube, globally optimum assignments were computed over 90% of the time in less than 10^6 movers. The adaptive schedules we have employed consistently outperform annealing implementations without adaptive schedules. The point at which it is desirable to change from adaptive to non-adaptive schedules is currently unknown.

We compared the performance of this mapping algorithm to the manually generated schedules produced in the design of a current real-time flight management system. The target architecture was a heterogeneous bus based system. Manual generation of schedules makes it possible to modify the software to change load distribution, inter-processor communication, etc. (i.e., effectively alter the weights on the graph) to meet operating constraints. As a result, the manually generated schedules were superior, but took on the order of hours to days to generate compared to seconds for the tools to return solutions. Furthermore, we were dealing with relatively few and large tasks creating high (e.g. 98%) system loads. As the number of tasks grows (due to decreasing granularity or more work) we expect the gap between automated mapping algorithms and manually generated solutions will close rapidly.

V. FUTURE WORK

This work is part of a larger effort to develop a multiprocessor system toolkit to support the conception, design, analysis, and development of large scale, medium-to-coarse grained multiprocessor architectures. Future research will expand the role of the mapping algorithm to consider its effect on reliability, and the feasibility of computing assignments that optimize other performance attributes such as response time.

- S. H. Bokhari, "On the Mapping Problem", *IEEE Transaction on Computers*, March 1981.
- S. Yalamanchili and D. T. Lee, "A Mapping Algorithm for Multiprocessor Architectures", 26th Allerton Conference on Computing, Communications, and Control, 1988.
- S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, "Optimization by Simulated Annealing", *Science*, vol. 220, no. 4598, May 1933, pp. 671-680.

Experience in Using SIMD and MIMD Parallelism for Computational Fluid Dynamics

Horst D. Simon*
Applied Research Branch
NASA Ames Research Center, Mail Stop T045-1
Moffett Field, CA 94035

Leonardo Dagum*
Applied Research Branch
NASA Ames Research Center, Mail Stop T045-1
Moffett Field, CA 94035

March 20, 1991

Abstract One of the key objectives of the Applied Research Branch in the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center is the accelerated introduction of highly parallel machines into a full operational environment. In this report we summarize some of the experiences with the parallel testbed machines at the NAS Applied Research Branch. We discuss the performance results obtained from the implementation of two Computational Fluid Dynamics (CFD) applications, an unstructured grid solver and a particle simulation, on the Connection Machine CM-2 and the Intel iPSC/860.

Keywords: parallel architectures, parallel algorithms.

AMS Subject Classification 76-08, 65W05, 65N99.

CR Subject Classification G.1.8, J.2, C.1.1, C.1.2.

1 Introduction

One of the key tasks of the Applied Research Branch in the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center is the accelerated introduction of highly parallel and related key hardware and software technologies into a full operational environment (see [1]). From 1988 1990 a testbed facility has been established for the development and demonstration of highly parallel computer technologies. Currently a 32k processor Connection Machine CM-2 and an 128 node Intel iPSC/860 are operated at the NAS Applied Research Branch. This testbed facility is envisioned to consist of successive generations of increasingly powerful highly parallel systems that are scalable to high performance capabilities beyond that of conventional supercomputers.

It is recognized within the scientific computing community that the most promising approach toward achieving very large improvements in computing performance is through the application of highly parallel architectures. To meet the future processing needs of the aerospace research community, the Applied Research Branch supports a research program aimed at achieving the best match of parallel processing technology to the most demanding research applications. In the last two years a number of large scale computational fluid dynamics applications have been implemented on the two testbed machines, and the potential of the parallel machines for production use has been evaluated. Beyond that, a systematic performance evaluation effort has been initiated (see [4]), and basic algorithm research has been continued.

In this report we will first give a brief description of the capabilities of the parallel machines at NASA Ames. Then we will discuss some of the research carried out in the implementation of Computational Fluid Dynamics (CFD) applications on these parallel machines. We focus here on those applications where we have more detailed knowledge because of our own involvement. an explicit 2D Euler solver for unstructured grids, and a simulation based on particle methods. Other applications based on structured grids will be mentioned briefly, as well as the NAS effort in parallel benchmarking. In a final section we offer some preliminary conclusions on the performance of current parallel machines for CFD applications, as well as the potential of the different architectures for production use in the future. Another summary of some of the results from NASA Ames is given by D. Bailey in [3].

2 Parallel Machines at NASA Ames

2.1 Connection Machine

The Thinking Machines Connection Machine Model CM-2 is a massively parallel SIMD computer consisting of many thousands of bit serial data processors under the direction of a front end computer. The system at NASA Ames consists of 32768 bit serial processors each with with 1 Mbit of memory and operating at 7 MHz. The processors and memory are packaged as 16 in a chip. Each chip also contains the routing circuitry which allows any processor to send and receive messages from any other processor in the system. In addition, there are 1024 64-bit Weitek floating point processors which are fed from the bit serial processors through a special purpose "Sprint" chip. There is one Sprint chip connecting every two CM chips to a Weitek. Each Weitek processor can execute an add and a multiply each clock cycle thus performing at 14 MFLOPS and yielding a peak aggregate performance of 14 GFLOPS for the system.

The Connection Machine can be viewed two ways, either as an 11-dimensional hypercube connecting the 2048 CM chips or a 10-dimensional hypercube connecting the 1024 processing elements. The first view is the "fieldwise" model of the machine which has existed since its introduction. This view admits to the existence of at least 32768 physical processors (when using the whole machine) each storing data in fields within its local memory. The second is the more recent "slicewise" model of the machine which admits to only 1024 processing elements (when using the whole machine) each storing data in slices of 32 bits distributed across the 32 physical processors in the processing element. Both models allow for "virtual processing", where the resources of a single processor or processing element may be divided to allow a greater number of virtual processors.

Regardless of the machine model, the architecture allows interprocessor communication to proceed in three manners. For very general communication with no regular pattern, the router determines the destination of messages at run time and directs the messages accordingly. This is referred to as general router communication. For communication with an irregular but static pattern, the message paths may be pre-compiled and the router will direct messages according to the pre-compiled paths. This is referred to as compiled communication and can be 5 times faster than general router communication. Finally, for communication which is perfectly regular and involves only shifts along grid axes, the system software optimizes the data layout by ensuring strictly nearest neighbor communication and uses its own pre-compiled paths. This is referred to as NEWS (for "NorthEastWestSouth") communication. Despite the name, NEWS communication is not restricted to 2 dimensional grids, and up to 31-dimensional NEWS grids may be specified. NEWS communication is the fastest.

The I/O subsystems connect to the data processors through an I/O controller. An I/O controller connects to 8192 processors through 256 I/O lines. There is one line for each chip but the controller can only connect to 256 lines simultaneously and must treat its 8k processors as two banks of 4k each. Each I/O controller allows transfer rates of up to 40 MB per second. In addition to an I/O controller there can be a frame buffer for color graphics output. Because it is connected directly to the backplane rather than through the I/O bus, the frame buffer can receive data from the CM processors at 256 MB per second. The system at NASA Ames has two frame buffers connected to two high resolution color monitors and four I/O controllers connected to a 20 GB DataVault mass storage system.

The Connection Machine's processors are used only to store data. The program instructions are stored on a front end computer which also carries out any scalar computations. Instructions are sequenced from the front end to the CM through one or more sequencers. Each sequencer broadcasts instructions to 8192 processors and can execute either independent of other sequencers or combined in two or four. There are two front end computers at NASA Ames, a Vax 8350 and a Sun 4/490, which currently support about 100 users. There are two sequencer interfaces on each computer which allow up to four concurrent users. In addition, the system software supports the Network Queue System (NQS) and time sharing through the CM Time Sharing System (CMTSS).

The Connection Machine system was first installed at NASA Ames in June of 1988. Since then the system has undergone a number of upgrades, the most recent being completed in February of 1991. An assessment of the system is given in [21]. Perhaps its greatest strength, from a user standpoint, is the robust system software. This is of critical importance to NASA as it moves its parallel machines into production mode.

2.2 Intel iPSC/860

The Intel iPSC/860 (also known as Touchstone Gamma System) is based on the new 64 bit i860 microprocessor by Intel. The i860 has over 1 million transistors and runs at 40 MHz. The theoretical peak speed is 80 MFLOPS in 32 bit floating point and 60 MFLOPS for 64 bit floating point operations. The i860 features 32 integer address registers, with 32 bits each, and 16 floating point registers with 64 bits each (or 32 floating point registers with 32 bits each). It also features an 8 kilobyte on-chip data cache and a 4 kilobyte instruction cache. There is a 128 bit data path between cache and registers. There is a 64 bit data path between main memory and registers.

The i860 has a number of advanced features to facilitate high execution rates. First of all, a number of important operations, including floating point add, multiply and fetch from main memory, are pipelined operations. This means that they are segmented into three stages, and in most cases a new operation can be initiated every 25 nanosecond clock period. Another advanced feature is the fact that multiple instructions can be executed in a single clock period. For example, a memory fetch, a floating add and a floating multiply can all be initiated in a single clock period.

A single node of the Touchstone Gamma system consists of the i860, 8 megabytes (MB) of dynamic random access memory, and hardware for communication to other nodes. For every 16 nodes, there is also a unit service module to facilitate access to the nodes for diagnostic purposes. The Touchstone Gamma system at NASA Ames consists of 128 computational nodes. The theoretical peak performance of this system is thus approximately 7.5 GFLOPS on 64 bit data.

The 128 nodes are arranged in a seven dimensional hypercube using the direct connect routing module and the hypercube interconnect technology of the iPSC/2. The point to point aggregate bandwidth of the interconnect system, which is 2.8 MB/sec per channel, is the same as on the iPSC/2. However the latency for the message passing is reduced from about 350 microseconds to about 90 microseconds. This reduction is mainly obtained through the increased speed of the i860 on the Touchstone Gamma machine, when compared to the Intel 386/387 on the iPSC/2. The improved latency is thus mainly a product of faster execution of the message passing software on the i860.

Attached to the 128 computational nodes of the NASA Ames system are ten I/O nodes, each of which can store approximately 700 MB. The total capacity of the I/O system is thus about 7 GB. These I/O nodes operate concurrently for high throughput rates. The complete system is controlled by a system resource module (SRM), which is based on an Intel 80386 processor. This system handles compilation and linking of source programs, as well as loading the executable code into the hypercube nodes and initiating execution. At present the SRM is a serious bottleneck in the system, due to its slowness in compiling and linking user codes. For example, the compilation of a moderate-sized application program often requires 30 minutes or more, even with no optimization options and no other users on the system.

During 1990 the iPSC/860 has been thoroughly investigated at NASA Ames. A first set of benchmark numbers, and some CFD applications performance numbers have been published in [2]. A more recent summary is given by Barszcz in [5]. As documented in [5] from an overall systems aspect the main bottleneck has been the SRM, which is not able to handle the demands of a moderately large user community (about 50 to 100 users) in a production environment. Another important result of the investigations was the outcome of a study by Lee [13]. Lee s analysis of the i860 floating point performance indicates that on typical CFD kernels the best performance to be expected is in the 10 MFLOPS range.

3 Structured Grid Applications

Structured grid codes, in particular multiblock structured grid codes, are one of the main production CFD tools at NASA Ames. A number of different efforts were directed toward the implementation of such capabilities on parallel machines. One of the first CFD results on the CM 2 was the work by Levit and Jespersen [15, 14], which was recently extended to three dimensions [16]. Their implementation is based on the successful ARC2D and ARC3D codes developed by Pulliam [20]. Work is in progress to implement F3D, a successor code to ARC3D, on the CM 2. On the iPSC/860 Weeratunga has implemented ARC2D (for early results see [2]), and work is in progress to implement F3D. Weeratunga also has developed a pseudo CFD application based on structured grids for the NAS Parallel Benchmark, which is described in chapter 3 of [4]. We will not discuss these efforts here in more detail and refer the interested reader to the references.

4 Unstructured Grid Applications

We discuss here work on an upwind finite-volume flow solver for the Euler equations in two dimensions that is well suited for massively parallel implementation. The mathematical formulation of this flow solver was proposed and implemented on the Cray-2 by Barth and Jespersen[6]. This solver has been implemented on the CM-2 by Hammond and Barth [11], and on the Intel iPSC/860 by Venkatakrishnan, Simon, and Barth [23].

The unstructured grid code developed by Barth is a vertex based finite volume scheme. The control volumes are non overlapping polygons which surround the vertices of the mesh, called the "dual" of the mesh. Associated with each edge of the original mesh is a dual edge. Fluxes are computed along each edge of the dual in an upwind fashion using an approximate Riemann solver Piecewise linear reconstruction is employed which yields second order accuracy in smooth regions. A 4 stage Runge Kutta scheme is used to advance the solution in time. Fluxes, gradients and control volumes are all constructed by looping over the edges of the original mesh. In the Cray implementation, vectorization is achieved by coloring the edges of the mesh.

It is assumed that a triangularization of the computational domain and the corresponding mesh has been computed. We will not present any more details here. A complete description of the algorithm can be found in [6, 11].

In both implementations the same test case has been used. The test case used is an unstructured mesh with 15606 vertices, 45878 edges, 30269 faces, 4 bodies, and 949 boundary edges. The flow was computed at a Mach number of .1 at 0 degrees angle of attack. The code for this test case runs at 150 Mflops on the NAS Cray-YMP at NASA Ames, and requires 0.39 seconds per time step.

4.1 SIMD Implementation of Unstructured Solver

For the implementation on the CM-2 Hammond and Barth [11] used a novel partitioning of the problem which minimizes the computation and communication costs on a massively parallel computer. In a mesh-vertex scheme, solution variables are associated with each vertex of the mesh and flux computation is performed at edges of the non-overlapping control volumes which surround each vertex. In conventional parallel implementations this operation is partitioned to be performed edge-wise, i.e., each *edge* of the control volume is assigned to one processor (edge-based). The resulting flux calculation contributes to two control volumes which share the particular edge.

In the partitioning used by Hammond and Barth, each *vertex* of the mesh is assigned to one processor (vertex-based). Flux computations are identical to the edge-based scheme but computed by processors associated with vertices. Each edge of the mesh joins a pair of vertices and is associated with one edge of the control volume.

One can direct edge (i,j) to determine which vertex in the pair computes the flux through the shared edge of the control volume, (k',j'). When there is a directed edge from i to j, then the processor holding vertex j sends its conserved values to the processor holding vertex i, and the flux across the common control volume edge is computed by processor i and accumulated locally. The flux through (k',j') computed by the processor holding vertex i is sent to the processor holding vertex j to be accumulated negatively. Hammond and Barth show that their vertex-based scheme requires 50% less communication and asymptotically identical amounts of computation as compared with the traditional edge-based approach.

Another important feature of the work by Hammond and Barth is the use of fast communication. A feature of the communication within the flow solver here is that the communication pattern, although irregular, remains static throughout the duration of the computation. The SIMD implementation takes advantage of this by using a mapping technique developed by Hammond and Schreiber [12] and a "Communication Compiler" developed for the CM-2 by Dahl [10]. The former is a highly parallel graph mapping algorithm that assigns vertices of the grid to processors in the computer such that the sum of the distances that messages travel is minimized. The latter is a software facility for scheduling irregular communications with a static pattern. The user specifies a list of source locations and destinations for messages which are then compiled into routing paths to be used at run time.

Hammond and Barth have incorporated the mapping algorithm and the communication compiler into the flow solver running on the CM-2 and have realized a factor of 30 reduction in communication time compared to using naive or random assignments of vertices to processors and the router. Using 8K processors of the CM-2 and a VP ratio of 2, Hammond and Barth carried out 100 time steps of the flow solver in about 71.62 seconds. This does not include setup time.

4.2 MIMD Implementation of Unstructured Solver

Similar to the SIMD implementation one of the key issues is the partitioning of the unstructured mesh. In order to partition the mesh Venkatakrishnan et al. [23] employ a new algorithm for the graph partitioning problem, which has been discussed recently by Simon [22], and which is based on the computation of eigenvectors of the Laplacian matrix of a graph associated with the mesh. Details on the theoretical foundations of this strategy can be found in [19]. Detailed investigations and comparisons to other strategies (cf. [22]) have shown that the spectral partitioning produces subdomains with the shortest boundary, and hence tends to minimize communication cost.

After the application of the partition algorithm of the previous section, the whole finite volume grid with triangular cells is partitioned into P subgrids, each subgrid contains a number of triangular cells which form a single connected region. Each subgrid is assigned to one proces sor. All connectivity information is precomputed, using sparse matrix type data structures.

Neighboring subgrids communicate to each other only through their interior boundary vertices which are shared by the processors containing the neighboring subgrids. In the serial version of the scheme, field quantities (mass, momentum and energy) are initialized and updated at each vertex of the triangular grid using the conservation law for the Euler equations applied to the dual cells. Each processor performs the same calculations on each subgrid as it would do on the whole grid in the case of a serial computation. The difference is that now each subgrid may contain both physical boundary edges and interior boundary edges, which have resulted from grid partitioning. Since a finite volume approach is adopted, the communication at the inter-processor boundaries consists of summing the local contributions to integrals such as volumes, fluxes, gradients etc.

The performance of the Intel iPSC/860 on the test problem is given in Table 1.

Table 1: Performance of Unstructured Grid Code on the Intel iPSC/860

Processors	secs/step	MFLOPS	efficiency(%)
2	7.58	7.7	83
4	3.82	15.3	83
8	2.01	29.1	79
16	1.11	52.7	71
32	0.61	95.9	65
64	0.33	177.3	60
128	0.21	278.6	47

5 Particle Methods

Particle methods of simulation are of interest primarily for high altitude, low density flows. When a gas becomes sufficiently rarefied the constitutive relations of the Navier-Stokes equations (i.e. the Stokes law for viscosity and the Fourier law for heat conduction) no longer apply and either higher order relations must be employed or the continuum approach must be abandoned and the molecular nature of the gas must be addressed explicitly. The latter approach leads to direct particle simulation.

In direct particle simulation, a gas is described by a collection of simulated molecules thus completely avoiding any need for differential equations explicitly describing the flow. By accurately modelling the microscopic state of the gas the macroscopic description is obtained through the appropriate integration. The primary disadvantage of this approach is that the computational cost is relatively large. Therefore, although the molecular description of a gas is accurate at all densities, a direct particle simulation is competitive only for low densities where accurate continuum descriptions are difficult to make.

For a small discrete time step, the molecular motion and collision terms of the Boltzmann equation may be decoupled. This allows the simulated particle flow to be considered in terms of two consecutive but distinct events in one time step, specifically there is a collisionless motion of all particles followed by a motionless collision of those pairs of particles which have been identified as colliding partners. The collisionless motion of particles is strictly deterministic and reversible. However, the collision of particles is treated on a probabilistic basis. The particles move through a grid of cells which serves to define the geometry, to identify colliding partners, and to sample the macroscopic quantities used to generate a solution.

The state of the system is updated on a per time step basis. A single time step is comprised of five events.

1. Collisionless motion of particles.
2. Enforcement of boundary conditions.
3. Pairing of collision partners.
4. Collision of selected collision partners.
5. Sampling for macroscopic flow quantities.

Detailed description of these algorithms may be found in [17] and [7]

5.1 SIMD Implementation of Particle Simulation

Particle simulation is distinct from other CFD applications in that there are two levels of parallel granularity in the method. There is a coarse level consisting of cells in the simulation (which are approximately equivalent to grid points in a continuum approach) and there is a fine level consisting of individual particles. At the time of the CM-2 implementation there existed only the fieldwise model of the machine, and it was natural for Dagum [7] to decompose the problem at the finest level of granularity. In this decomposition, the data for each particle is stored in an individual virtual processor in the machine. A separate set of virtual processors (or VP set) stores the geometry and yet another set of virtual processors stores the sampled macroscopic quantities.

This decomposition is conceptually pleasing however in practice the relative slowness of the Connection Machine router can prove to be a bottleneck in the application. Dagum [7] introduces several novel algorithms to minimize the amount of communication and improve the overall performance in such a decomposition. In particular, steps 2 and 3 of the particle simulation algorithm requre a somewhat less than straightforward approach.

The enforcement of boundary conditions requires particles which are about to interact with a boundary to get the appropriate boundary information from the VP set storing the geometry data. Since the number of particles undergoing boundary interaction is relatively small, a master/slave algorithm is used to minimize both communication and computation. In this algorithm, the master is the VP set storing the particle data. The master creates a slave VP set large enough to accommodate all the particles which must undergo boundary interactions. Since the slave is much smaller than the master, instructions on the slave VP set execute much faster. This more than makes up for the time that the slave requires to get the geometry information and to both get and return the particle information.

The pairing of collision partners requires sorting the particle data such that particles occupying the same cell are represented by neighboring virtual processors in the one dimensional NEWS grid storing this data. Dagum [8] describes different sorting algorithms suitable for this purpose. The fastest of these makes use of the realization that the particle data moves through the CM processors in a manner analogous to the motion of the particles in the simulation. The mechanism for disorder is the motion of particles, and the extent of motion of particles, over a single time step, is small. This can be used to tremendously reduce the amount of communication necessary to re-order the particles.

These algorithms have been implemented in a two-dimensional particle simulation running on the CM-2. At the time of implementation, the CM-2 at NASA Ames had only 64k bits of memory per processor which was insufficient to warrant a three-dimensional implementation. Furthermore, the slicewise model of the machine did not exist and the machine had the slower 32-bit Weitek's which did not carry out any integer arithmetic. Nonetheless, with this smaller amount of memory and fieldwise implementation, the code was capable of simulating over 2.0×10^6 particles in a grid with 6.0×10^4 at a rate of 2.0μsec/particle/timestep using all 32k processors (see [7]). By comparison, a fully vectorized equivalent simulation on a single processor of the Cray YMP runs at 1.0μsec/particle/timestep and 86 MFLOPS as measured by the Cray hardware performance monitor. (Note that a significant fraction of a particle simulation involves integer arithmetic and the MFLOP measure is not completely indicative of the amount of computation involved). Currently, work is being carried out to extend the simulation to three dimensions using a parallel decomposition which takes full advantage of the slicewise model of the machine.

5.2 MIMD Implementation of Particle Simulation

The MIMD implementation differs from the SIMD implementation not so much because of the difference in programming models but because of the difference in granularity between the machine models. Whereas the CM 2 has 32768 processors, the iPSC/860 has only 128. Therefore on the iPSC/860 it is natural to apply a spatial domain decomposition rather than the data object decomposition used on the CM-2.

Table 2: Performance of Particle Simulation on the Intel iPSC/860

Processors	μs/prt/step	MFLOPS	efficiency(%)
2	24.4	3.5	97
4	12.5	6.9	95
8	6.35	13.5	93
16	3.25	26.5	91
32	1.63	52.8	91
64	0.85	101	87
128	0.42	215	88

In McDonald's [18] implementation, the spatial domain of the simulation is divided into a number of sub-domains or regions equal to the desired number of node processes. Communication between processes occurs as a particle passes from one region to another and is carried out asynchronously, thus allowing overlapping communication and computation. Particles crossing region "seams" are treated simply as an additional type of boundary condition. Each simulated region of space is surrounded by a shell of extra cells that, when entered by a particle, directs that particle to the neighboring region. This allows the representation of simulated space (i.e. the geometry definition) to be distributed along with the particles. The aim is to avoid maintaining a representation of all simulated space which, if stored on a single processor, would quickly become a serious bottleneck for large simulations, and if replicated would simply be too wasteful of memory.

Within each region the sequential or vectorized particle simulation is applied. This decomposition allows for great flexibility in the physical models that are implemented since node processes are asynchronous and largely independent of each other. Recall that communication between processes is required only when particles cross region seams. This is very fortuitous since the particle motion is straightforward and fully agreed upon. The important area of research has to do with the modelling of particles, and since this part of the problem does not directly affect communication, particle models can evolve without requiring great algorithmic changes.

McDonald's implementation is fully three-dimensional. The performance of the code on a 3D heat bath is given in Table 2.

At the present time the domain decomposition is static, however work is being carried out to allow dynamic domain decomposition thus permitting a good load balance to exist throughout a calculation. The geometry and spatial decomposition of the heat bath simulation *exaggerated* the area to volume ratio of the regions in order to more closely approximate the communication expected in a real application with dynamic load balancing. The most promising feature of these results is the linear speed up obtained, indicating that the performance of the code should continue to increase with increasing numbers of processors.

6 Conclusions

On the unstructured grid code the performance figures are summarized in Table 3, where all MFLOPS numbers are Cray Y-MP equivalent numbers.

Table 3 Performance Comparison of Unstructured Grid Code

Machine	Processors	secs/step	MFLOPS
Cray Y-MP	1	0.39	150.0
Intel iPSC/860	64	0.33	177.3
	128	0.21	278.6
CM-2 (32 bit)	8192	0.72	81.3

For the particle methods the corresponding summary of performance figures can be found in Table 4. The figures in Table 4 should be interpreted very carefully. The simulations run on the different machines were comparable, but not identical. The MFLOPS are Cray Y-MP equivalent MFLOPS ratings based on the hardware performance monitor.

Table 4: Performance Comparison of Particle Simulation Code

Machine	Processors	μsecs/particle/step	MFLOPS
Cray 2	1	2.0	43
Cray Y-MP	1	1.0	86
Intel iPSC/860	128	0.4	215
CM-2 (32-bit)	32768	2.0	43

The results in Tables 3 and 4 demonstrate a number of points. Both unstructured grid computations and the particle simulations are applications which a priori are not immediately parallelized, and for which both on SIMD and MIMD machines considerable effort must be expended in order to obtain an efficient implementation. It has been demonstrated by the results obtained at NASA Ames that this can be done, and that supercomputer level performance can be obtained on current generation parallel machines. Furthermore the particle simulation code on the CM-2 is a production code currently used to obtain production results (see [9]). The iPSC/860 implementation should be in production use by the end of 1991.

Our results also demonstrate another feature which has been found across a number of applications at NASA Ames: massively parallel machines quite often obtain only a fraction of their peak performance on realistic applications. In the applications considered here, the requirement for unstructured, general communication has been the primary impediment in obtaining the peak realizable performance from these machines. Neither the CM-2 nor the the iPSC/860 deliver the communication bandwidth necessary for these CFD applications. This situation is even worse for implicit algorithms (see e.g. [2]). Experience has shown that CFD applications require on the order of one memory reference per floating point operation and a balanced system should have a memory bandwidth comparable to its floating point performance. In these terms, current parallel systems deliver only a fraction of the required bandwidth.

Acknowledgement. We wish to thank our colleagues with the NAS Applied Research Branch, whose work has been discussed here: D. Bailey, E. Barszcz, R. Fatoohi, T. Lasinski, C. Levit, V. Venkatakrishnan, and S. Weeratunga. We also thank T. Barth, D. Jespersen, J. McDonald, (all NASA Ames) and P. Fredericksen, S. Hammond, and R. Schreiber at RIACS for their contributions to this summary report.

References

[1] *Numerical Aerodynamic Simulation Program Plan.* NAS Systems Division, NASA Ames Research Center, October 1988.

[2] D. Bailey, E. Barszcz, R. Fatoohi, H. Simon, and S. Weeratunga. Performance results on the intel touchstone gamma prototype. In David W. Walker and Quentin F. Stout, editors, *Proceedings of the Fifth Distributed Memory Computing Conference*, pages 1236 - 1246, IEEE Computer Society Press, Los Alamitos, California, 1990.

[3] D. H. Bailey. *Experience with Parallel Computers at NASA Ames.* Technical Report RNR-91-07, NASA Ames Research Center, Moffett Field, CA 94035, February 1991.

[4] D. H. Bailey, J. Barton, T. Lasinski, and H. Simon. *The NAS Parallel Benchmarks.* Technical Report RNR-91-02, NASA Ames Research Center, Moffett Field, CA 94035, January 1991.

[5] E. Barszcz. *One Year with an iPSC/860.* Technical Report RNR-91-01, NASA Ames Research Center, Moffett Field, CA 94035, January 1991.

[6] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *Proceedings, 27th Aerospace Sciences Meeting*, January 1989. Paper AIAA 89-0366.

[7] L. Dagum. *On the Suitability of the Connection Machine for Direct Particle Simulation.* Technical Report 90.26, RIACS, NASA Ames Research Center, Moffett Field, CA 94035, June 1990.

[8] L. Dagum. Sorting for particle flow simulation on the connection machine. In Horst D. Simon, editor, *Research Directions in Parallel CFD*, MIT Press, Cambridge(to appear), 1991.

[9] L. Dagum. *Lip Leakage Flow Simulation for the Gravity Probe B Gas Spinup Using PSiCM.* Technical Report RNR-91-10, NASA Ames Research Center, Moffett Field, CA 94035, March 1991.

[10] E. Denning Dahl. Mapping and compiled communication on the connection machine system. In David W. Walker and Quentin F. Stout, editors, *Proceedings of the Fifth Distributed Memory Computing Conference*, pages 756 – 766, IEEE Computer Society Press, Los Alamitos, California, 1990.

[11] S. Hammond and T.J. Barth. On a massively parallel Euler solver for unstructured grids. In Horst D. Simon, editor, *Research Directions in Parallel CFD*, MIT Press, Cambridge(to appear), 1991.

[12] S. Hammond and R. Schreiber. *Mapping Unstructured Grid Problems to the Connection Machine.* Technical Report 90.22, RIACS, NASA Ames Research Center, Moffett Field, CA 94035, October 1990.

[13] K. Lee. *On the Floating Point Performance of the i860 Microprocessor.* Technical Report RNR-90-019, NASA Ames Research Center, Moffett Field, CA 94035, 1990.

[14] C. Levit and D. Jespersen. A computational fluid dynamics algorithm on a massively parallel computer. *Int. J. Supercomputer Appl.*, 3(4):9 - 27, 1989.

[15] C. Levit and D. Jespersen. *Explicit and Implicit Solution of the Navier-Stokes Equations on a Massively Parallel Computer.* Technical Report, NASA Ames Research Center, Moffett Field, CA, 1988.

[16] C. Levit and D. Jespersen. *Numerical Simulation of a Flow Past A Tapered Cylinder.* Technical Report RNR-90-20, NASA Ames Research Center, Moffett Field, CA 94035, October 1990.

[17] J. D. McDonald. *A Computationally Efficient Particle Simulation Method Suited to Vector Computer Architectures.* PhD thesis, Stanford University, Dept. of Aeronautics and Astronautics, Stanford CA 94305, December 1989.

[18] J. D. McDonald. *Particle Simulation in a Multiprocessor Environment.* Technical Report RNR-91-02, NASA Ames Research Center, Moffett Field, CA 94035, January 1991.

[19] A. Pothen, H. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Mat. Anal. Appl.*, 11(3):430 – 452, 1990.

[20] T. H. Pulliam. Efficient solution methods for the Navier-Stokes equations. 1986. Lecture Notes for The Von Karman Institute for Fluid Dynamics Lecture Series, Jan. 20 - 24.

[21] R. Schreiber. *An Assessment of the Connection Machine.* Technical Report 90.40, RIACS, NASA Ames Research Center, Moffett Field, CA 94035, June 1990.

[22] H. D. Simon. *Partitioning of Unstructured Problems for Parallel Processing.* Technical Report RNR-91-08, NASA Ames Research Center, Moffett Field, CA 94035, February 1991. (to appear in Computing Systems in Engineering).

[23] V. Venkatakrishnan, H. Simon, and T. Barth. *A MIMD Implementation of a Parallel Euler Solver for Unstructured Grids.* Technical Report RNR-91-xx, NASA Ames Research Center, Moffett Field, CA 94035, 1991. (in preparation).

AUTOMATIC PARTITIONING OF FINITE ELEMENT/FINITE DIFFERENCE MESHES FOR PARALLEL PROCESSING

Charbel Farhat
Department of Aerospace Engineering
And Center for Space Structures and Controls
University of Colorado
Boulder, CO 80309-0429 (U. S. A.)

Abstract

The various forms of parallel numerical algorithms that speed up finite element computations are as different as the number of researchers working on the problem. However, most of the recently proposed concurrent computational strategies stem from the "divide and conquer" paradigm and require domain decomposition or mesh partitioning. In this talk, we present and discuss a family of simple and efficient non-numerical algorithms for the automatic decomposition of arbitrary finite element and finite difference meshes into a specified number of adequatly connected and load balanced submeshes. These algorithms cover a wide spectrum of parallel architectures (local memory, shared memory, massively parallel) and feature a large range of approaches (greedy algorithms, projection methods, bandwidth minimization). In particular we address the issues of communication bandwidth [2, 8], interface bottleneck [3] and element recursion [5]. We show how these various decomposers can be applied to implement parallel, massively parallel, and parallel/vector explicit and implicit direct, frontal and iterative finite element and finite difference algorithms, and report on their performance results on the iPSC/2-32, Connection Machine CM2, and CRAY Y-MP- 8. For local memory multiprocessors, each partitioning scheme is augmented with a mapping algorithm [1, 7, 4] which attempts to assign directly connected submeshes to directly interconnected processors.

The proposed mesh decomposers are packaged into a software tool which is hoped to relieve the burden of the preprocessing phase from the methods developers.

References

[1] Bokhari S. H., "On the Mapping Problem," *IEEE Trans. Comp.*, Vol. C-30, No. 3, (1981) pp. 207-214.

[2] Farhat C. and E. Wilson, "A New Finite Element Concurrent Computer Program Architecture," *Int. J. Num. Meth. Eng.*, Vol. 24, No. 9, (1987) pp. 1771-1792.

[3] Farhat C., 'A Simple and Efficient Automatic FEM Domain Decomposer," *Comp. & Struc.*, Vol. 28, No. 5, (1988) pp. 579-602.

[4] Farhat C., "On the Mapping of Massively Parallel Processors Onto Finite Element Graphs," *Comp. & Struc.*, Vol. 32, No. 2, (1989) pp. 347-354.

[5] Farhat C. and L. Crivelli, A General Approach to Nonlinear FE Computations on Shared Memory Multiprocessors," *Comp. Meth. Appl. Mech. Eng.*, Vol. 72, No. 2, (1989) pp. 153-172.

[6] Farhat C., N. Sobh and K. C. Park, "Dynamic Finite Element Simulations on the Connection Machine," *Int. J. High Speed Comp.*, Vol. 1, No. 2, (1989) pp. 289-302.

[7] Flower J. W., S. W. Otto, and M. C. Salama, "A Preprocessor for Irregular Finite Element Problems," CalTech/JPL Report C3P-292, July 1986.

[8] Malone J. G., "Automated Mesh Decomposition and Concurrent Finite Element Analysis for Hypercube Multiprocessors Computers," *Comp. Meth. Appl. Mech. Eng.*, Vol. 70, (1988) No. 1, pp. 27-58.

NON-LINEAR ELASTICITY SOLVED BY A DOMAIN-DECOMPOSITION METHOD ON A HYPERCUBE

Yann-Hervé De Roeck
CERFACS,
Toulouse, France

Abstract : We want to compute the equilibrium positions of hyperelastic bodies under large strain using parallel machines with distributed memory architecture, namely a hypercube. We achieve this goal by using a domain-decomposition method at the level of each linearized problem. Assigning one sub-domain per node, the computation and assembly of the local stiffness matrices then become independent tasks, avoiding any communication between the nodes. The remaining computation on the linearised problem involves a mixed solver, meaning that we use both a local direct solver and a global iterative scheme. Namely, a preconditioned conjugate gradient is used at the interface, with a preconditioner that has been chosen in sight of keeping the high granularity of the parallelism. The target machine of our experiments is an Intel iPSC/2 hypercube 32SX.

1- The discretized non-linear problem :

For the modelization of bodies that can undergo large deformations, we choose the Lagrangian formulation. This means that all variables are defined and maintained in a reference configuration, the main unknown being the displacement field $u(x)$ of each particle of the domain Ω once the loading has been applied.

The equilibrium equation, once discretized on a Finite Elements basis $\{\varphi_\alpha\}$, is solved by using a Newton-type method. For *compressible* materials, the constitutive law takes such a form that each step can be described as :

$\boxed{\text{Step } n \rightarrow n+1}$

$$\begin{aligned}
\mathrm{u}^{n+1} &= \mathrm{u}^{n} - \left[\mathrm{A}^{n}\right]^{-1}\left(\mathrm{G}^{n} - \mathrm{H}^{0}\right) \ , \\
\text{with} \quad \mathrm{A}^{n}_{\alpha\beta} &= \int_\Omega \nabla\varphi_\beta^T : \frac{\partial^2 W}{\partial F^2}(u^n) : \nabla\varphi_\alpha \, dx \ , \\
\mathrm{G}^{n}_{\alpha} &= \int_\Omega \frac{\partial W}{\partial F}(u^n) : \nabla\varphi_\alpha \, dx \ , \\
\mathrm{H}^{0}_{\alpha} &= \int_\Omega f.\varphi_\alpha \, dx + \int_{\partial\Omega} g.\varphi_\alpha \, da \ , \text{ the load } , \\
\text{where } F(x) &= Id + \nabla u(x) \text{ is the deformation gradient } , \\
\text{and } W(F) & \quad \text{is the specific internal elastic energy } .
\end{aligned}$$

In turn, for *isotropic* materials, W only depends on the invariants of the right Cauchy-Green tensor F^TF :

$$\begin{aligned}
I_1 &= \mathrm{Tr}(F^TF) \ , \\
I_2 &= \mathrm{Tr}((\mathrm{adj}F)^T\mathrm{adj}F) \ , \\
J &= \det(F) \ .
\end{aligned}$$

A typical example is given by the following law for hyperelastic materials :

$$W(F) = C_1(I_1-3)+C_2(I_2-3)+a(J^2-1)-(2C_1+4C_2+2a)\log J \ ,$$

where C_1, C_2 and a are experimental constants.

To enhance the Newton scheme, we have also applied the following numerical features : the incremental loading and the arc-length continuation, which enable to follow awkward problems like buckling (one of the major needs of large deformations modelizations).

2- The mixed linear solver.

At each iteration of this process, we must solve a linear system of equations of the kind.

$$\mathrm{A}^n(\mathrm{U}^n) \, . \, \mathrm{U}^{n+1} = \mathrm{F}^n(\mathrm{U}^n) \ .$$

The costly steps of constructing, assembling and factorizing A^n require on a distributed memory machine a coarse-grained parallelism, in order to avoid overwhelming communications. Therefore, the domain decomposition methods achieves this goal by subdividing *ab initio* the geometrical reference domain Ω into non-overlapping subdomains Ω_i, separated by an interface Γ. Consequently, the computation and assembly of these subdomain matrices A_i become fully independent tasks.

As we could have stated in the previous paragraph, the global stiffness matrix A is symmetric, positive, and most often definite (SPD). It is composed from the contributions of the following submatrices:

$\mathring{\mathrm{A}}_i$. traces the degrees of freedom internal to Ω_i,
B_i : interaction between Ω_i and Γ,
$\bar{\mathrm{A}}_i$: contributions due to the elements of Ω_i on Γ.

By performing Gaussian elimination over the degrees of freedom related to the interior of the subdomains Ω_i, one obtains a linear system whose unknowns are the degrees of freedom on the interface and whose matrix is the Schur complement matrix S on the interface Γ. Then, omitting the projections and the mappings, one might observe its useful additive property:

$$\mathrm{S} = \sum_i \mathrm{S}_i = \bar{\mathrm{A}}_i - \mathrm{B}_i^T \mathring{\mathrm{A}}_i^{-1} \mathrm{B}_i \ .$$

All components of the local Schur complement matrices can be computed, as in the well known *substructuring technique*. However, a much less expensive approach consists in keeping an *implicit* definition of the local Schur complement, and solving the interface problem using a *Conjugate Gradient* method (CG), because S in turn is also SPD. At step k of the CG, one has to form in parallel the implicit matrix-vector products $\mathrm{S}p_k$ where p_k is the k^{th} descent direction, which requires the solution of a linear system of matrix $\mathring{\mathrm{A}}_i$.

Based on an LDL^T decomposition of $\mathring{\mathrm{A}}_i$, a direct solver is used in each domain. This factorization is performed *once and for all* during the initialization step of the inner CG loop.

However, this method requires a preconditioning technique to be efficient. Benefitting from our experiments with the linear three-dimensional elasticity, we have chosen the preconditioner proposed in an analytical form by Glowinski *et al* in [1]. This amounts to approximating

$$\begin{aligned}
\mathrm{S}^{-1} &= \left(\sum_i \mathrm{S}_i\right)^{-1} \\
\text{by } \mathrm{M}^{-1} &= \sum_i \mathrm{D}_i \mathrm{S}_i^{-1} \mathrm{D}_i^T \ .
\end{aligned}$$

where D_i are diagonal weighting matrices. This preconditioner is often referred as the *Neumann* preconditioner, because of the analytical boundary conditions described at the interface. Similarly, using the Schur complement amounts to treat a *Dirichlet* problem regarding the interface.

Without computing the local Schur Complement S_i explicitly, there exists an *implicit* definition of their inverses .

$$S_i^{-1} = \begin{pmatrix} 0 & 1_{|\Gamma_i} \end{pmatrix} \begin{pmatrix} \mathring{A}_i & B_i \\ B_i^T & \bar{A}_i \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1_{|\Gamma_i} \end{pmatrix}.$$

Thus, only an LDL^T factorization of the local stiffness matrices has to be performed, in order to compute products of vectors by the preconditioning matrix M.

In the linear framework, we have already extensively studied this method, in [2]. It shows a great potential for parallelism, especially for distributed memory-architecture and it proved to be robust in problems arising from anisotropic and not-homogeneous materials. We also have computed theoretical bounds for the condition number of the iteration matrix. in [3], we proved that in a general partitioning, it grows like $O\left(\frac{1}{d^2}\left(1+\log(\frac{d}{h})\right)\right)$, d being the average diameter of a subdomain and h the scale of the mesh. The weak dependency over h means that the preconditioner is still profitable when the mesh is refined. However, the number of subdomains should not increase too much, and a limit of 16 seems reasonable, especially when the partitioning is performed in the three-dimensional space.

Without the optimality of the preconditioner described in [5] by Smith, whose condition number is independent of d and h, the "Neumann" preconditioner remains local and easy to construct on a large unstructured mesh: indeed it requires neither the explicit Schur Complement matrices nor any coarse mesh.

This versatility of the "Neumann" preconditioner favors its implementation in non-linear Finite Element problem.

<u>The data structure for the interface :</u> The distributed architecture of the computer must be taken into account in the choice of a suitable data structure for the interface.

At each iteration of the PCG, there are 3 interface vectors to be stored: u_k, r_k and p_k the displacement, the residual and the descent direction, respectively; and 2 interface vectors to be used $t = \sum_i S_i p_k$ and $z = Mr_k$ the conjugate direction and the preconditioned residual, respectively.

We describe a strategy of implementation, called *global interface*, where each node of the machine redundantly stores the whole interface, as opposed to the so-called *local interface*, where each node of the machine only knows the interface degrees of freedom which are in its immediate neighborhood.

This *global interface* also induces some redundant computation, but reduces the number of communications. On the one hand, the vectors to be transferred are longer, on the other hand, no communications are needed to perform the dot-products (whereas in the other approach, local weighted subproducts are computed and then gathered).

With this data structure, the only exchanges are the global summations: those are scheduled to be performed by physical directions of links (see Saad [4]), thus no special mapping of the subdomains onto the nodes is required, e.g. by a binary Gray-Code. Thanks to the unicity of the structure of the interface, one level of indirect addressing is also suppressed at the gathering operation.

A complex splitting becomes more difficult to handle with the global data structure. However, with no complementary communications, the code has been implemented with the following feature. two nodes are dedicated to each subdomain, one storing and using the *Dirichlet* solver, the other one taking care of the *Neumann* solver. Notice that on a distributed-memory machine, this approach leads to a good parallelization of the memory management and postpones the risk of lack of memory for bigger problems. Of course, these two solvers are still accessed sequentially, but it implies that for a given problem, fitted to n processors in memory requirements, one splits the body into $\frac{n}{2}$ subdomains, overcoming some difficulties of convergence previously quoted. Examples show that it is a matter of trade off.

The results that will be shown have been computed on the Intel iPSC/2 32SX of ONERA, thanks to a cooperation between the Groupe de Calcul Parallèle of ONERA and the Parallel Algorithms Group at CERFACS.

References

[1] J.-F. Bourgat, R. Glowinski, P. Le Tallec and M. Vidrascu, *Variational formulation and algorithm for trace operator in domain decomposition calculations* in T. Chan, R. Glowinski, J. Periaux and O. Widlund, Eds., *Proceedings of the second international symposium on domain decomposition methods, Los Angeles, California, January 14-16, 1988*, SIAM, Philadelphia, 1989.

[2] P. Le Tallec, Y.-H. De Roeck and M. Vidrascu, *Domain decomposition methods for large linearly elliptic three dimensional problems.*, to appear in: J. of Computational and Applied Mathematics 31 (1991), 93 117 Elsevier Science Publishers, Amsterdam.

[3] Y.-H. De Roeck and P. Le Tallec, *Analysis and test of a local domain-decomposition preconditioner*, to appear in *Proceedings of the fourth international symposium on domain decomposition methods, Moscow, USSR, May 1990*. SIAM, Philadelphia, 1991.

[4] Y. Saad and M. H. Schultz, *Data Communication in Hypercubes*, Journal of Parallel and Distributed Computing 5, 000-000 (1988)

[5] B. Smith, *An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems* Technical Report 182, department of Computer Science, Courant Institute, 1989.

Numerical treatment of integral equations on iPSC

Armel de La Bourdonnaye
ONERA, Parallel Computing Division
B.P.72 , 92322 Chatillon Cedex, France

March 5, 1991

Abstract

We are dealing with integral equations related to Helmholtz equation. They come from scattering problems around a compact objet. The usual way of discretizing them leads to a full complex non-hermitian matrix. This drastically limits the size of computable problems because of the limited size of memory on computers .

1 Preconditioning

The algorithm used for resolution is Generalized Conjugate Residual Algorithm. It needs to store all directions of descent. Preconditioning has two advantages. First it reduces the CPU time for the resolution and second it leads to fewer directions to be stored, so that we can test meshes with more points and so higher frequencies.

The preconditioning matrix we will focus on consists of a subpart of the full matrix. More precisely we will only retain coefficients that come from interaction between points near from each other (near means the distance is less than a few wavelengths).
We first show some theoretical results. The point is that, in a certain extent, we can explane how that preconditioner acts. Indeed it tends to diminish the highest eigenvalues of the matrix with no precise effect on the lower ones. What is really interesting is that these results remain true when frequency grows to infinity.

We then present two series of numerical issues of that preconditioned algorithm. The first one aims to illustrate the reduction of the number of iterations. We can see in tables 1 and 2 that the most remarquable facts is that the number of iterations is reduced from typically 10 to 1 and that seems not to vary [illegible] the frequency. Table 1 shows numbers of iterations for the unpreconditioned algorithm for various frequencies. In these tests, the scatterer is a sphere of radius 1, with sound speed equals to 333. The number of degrees of freedom is 1026. The incident wave is spherical harmonic. In table 2 is we can see the number of iterations needed to achieve convergence. The parameter δ is the maximal distance between 2 points whose mutual interaction is taken into account in the preconditioning matrix. The

ϵ	[illegible]Hz	500Hz	600Hz
[illegible]	90	> 300	185
10[illegible]	> 300	> 300	> 300
10[illegible]	> 300	> 300	> 300
ϵ	700Hz	750Hz	780Hz
[illegible]	85	265	>300
10^{-5}	> 300	> 300	> 300
10^{-6}	> 300	> 300	>300

Table 1: Unpreconditioned GCRA.

δ	780Hz			750Hz		
	10^{-4}	10^{-5}	10^{-6}	10^{-4}	10^{-5}	10^{-6}
0.40m	3	5	22	2	4	26
0.36m	2	5	35	2	4	25
0.32m	3	5	21	2	4	12
0.28m	3	8	20	2	7	17
0.24m	2	30	> 50	2	17	> 50
δ	700Hz			600Hz		
	10^{-4}	10^{-5}	10^{-6}	10^{-4}	10^{-5}	10^{-6}
0.40m	3	4	10	2	4	10
0.36m	3	4	9	3	7	11
0.32m	3	4	10	3	10	18
0.28m	3	11	24	5	27	> 50
0.24m	4	15	49	4	17	45

Table 2: Preconditioned GCRA.

second series of results presents comparisons between a monoprocessor (CrayII) and a parallel machine (iPSC-2) in terms of CPU time. in table 3 we present tests for two sizes of mesh. The scatterer is the same as before and the incident wave is a plane one. In that table, resolution time is the sum of the time of assembly of the matrix and the time of the GCRA.

Mesh n^o1 : 468 points and 968 triangles.

Mesh n^o2 : 1026 points and 2048 triangles.

We can see that a hypercube with 32 nodes is about the fifth of a processor of CRAYII. Next we will study the way of parallelizing some crucial pieces of the code on the iPSC-2. We will see what we can expect in terms of speed-up related to local granularity of the calculus. We will see that for a dot product we must increase this granularity to maintain efficiency as we increase the number of nodes when for a matrix-vector product it can remain constant if we take care. In table 4 we present actual times of computation for a matrix-vector product. $Nloc$ is the size of a vector divided by the number of processor used. T_{cal} and T_{comm} are respectively the time due to computation and communications.

Test	Assembly		Iteration		Resolution	
	C-II	5-Cube	C-II	5-Cube	C-II	5-Cube
n^o1	30	235,6	0,95	6,3	100	265
n^o2	263	021	2	17	267	1145

Table 3. Compared times. Cray-II vs n-Cube (in seconds).

nloc	256		128	
	T_{cal}	T_{comm}	T_{cal}	T_{comm}
5-cube	*	*	10072	53
4-cube	*	*	5032	28
3-cube	10063	26	2516	16
2-cube	5039	13	1259	9
nloc	**64**		**32**	
	T_{cal}	T_{comm}	T_{cal}	T_{comm}
5-cube	2516	21	630	20
4-cube	1260	17	319	13
3-cube	634	11	157	8
2-cube	315	6	78	5

Table 4. Matrix-vector product (time in ms).

Parallel Grid and Multigrid Methods for Distributed Memory Architectures*

Karl Solchenbach

PALLAS GmbH, Hohe Str. 73, D-5300 Bonn

1 Introduction

Algorithms for the numerical solution of PDEs are typically based on grid data structures (either regular or irregular ones). These algorithms are characterized by inherent parallelism and locality: values at different grid points can be calculated simultaneously and the - usually iterative - calculation of a grid-point value involves only values at certain neighboring points.

Due to these properties grid based algorithms (like red-black relaxation) can be implemented on distributed memory architecures very efficiently. Grids are decomposed into subgrids (grid partitioning) and each subgrid is connected to a processor. The locality of the grid operator guarantees that the communication between processors is limited.

The implementation of standard multigrid methods is also based on the grid partitioning approach. The parallel efficiency of multigrid, however, is somewhat less than that of the corresponding single grid method. This is mainly due to the high communication/calculation ratio and the short message length on coarse grids. Nevertheless, the numerical efficiency of multigrid outperforms these losses easily.

2 Parallel grid-based applications

The mathematical model of many different supercomputer applications are formulated as (systems of) partial differential equations (PDEs). The discretization of the PDEs most naturally leads to a grid based formulation of the problem, i.e. grid data structures and grid-based algorithms.

The implementation on distributed memory parallel computers requires

- the parallelization of the existing algorithms or their substitution by new parallel algorithms;
- the distribution of the data structure to the local memory units. The data distribution should try to preserve *locality* (i.e. minimize communication) and to achieve *load balancing*.

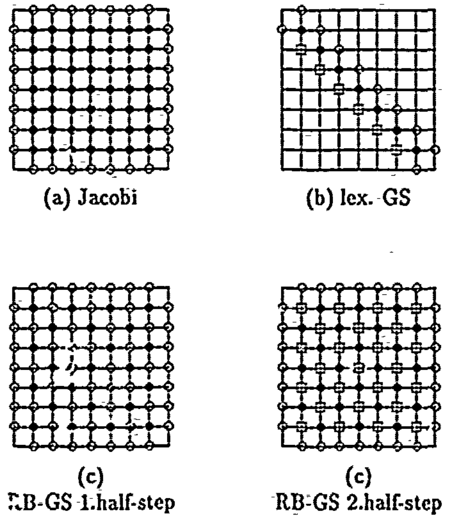
2.1 Grid data structures

Distribution strategies and tools have been developed for two classes of grid structures:

Regular grids are characterized by direct addressing of the grid points and a rectangular or cuboid address space. Geometrical neighbors are also logical neighbors.
Block-structured grids are composed of several regular grids. Each single block shows internally a regular grid structure; the block structure itself, however, is irregular (with certain restrictions).

Meanwhile also codes based on *irregular* grids (as used by Finite Element methods) and *locally refined* grids have been implemented on distributed memory parallel computers. Efficient and and comfortable tools for these structures, however, have to be developed yet.

2.2 Parallel grid algorithms



(a) Jacobi

(b) lex. GS

(c) RB-GS 1.half-step

(c) RB-GS 2.half-step

Figure 1: Jacobi and Gauss Seidel relaxation schemes. • denotes grid points which can be calculated independently in parallel, ○ denotes grid points with old values, and □ denotes grid points with already calculated new values.

A grid algorithm is a (usually iterative) method which calculates the value of a grid function at one point as a function of values defined at neighboring points. The iteration (also called relaxation) can be characterized as *Jacobi*-type (the new iterate at a grid point is calculated using only old neighboring values) or *Gauss-Seidel*-type (using already calculated new neighboring values). Obviously, Jacobi-type methods are completely parallel since the calculation in each grid point can be performed independently (cf. Figure 1 (a)). If the number of grid points is N the parallelism is also N.

The parallelism of Gauss-Seidel methods depends on the order in which the grid points are processed. Lexicographic ordering implies that only points on diagonal lines can be calculated in parallel (cf. Figure 1 (b)).

For Gauss-Seidel methods, a far better degree of parallelism, namely $N/2$, is obtained by "coloring" the grid points appropriately and processing all points of the same color simultaneously, e.g. the so-called red-black (RB-) relaxation (cf. Figure 1 (c)).

2.3 Grid partitioning

The usual way to implement parallel grid algorithms on a distributed memory system is based on the method of *grid partitioning*. The computational domain (grid) is divided into several subgrids which are assigned to parallel *processes*.

Each relaxation step can be performed on a subset of interior points of the subgrid (• in Figure 2) independently. Calculation of values at interior boundary points (○ in Figure 2), however, needs the values from neighboring subgrids (=processes). Since the processes have no common data space these values somehow have to be made available. Instead of transferring the values individually at the time they are needed it is more efficient to have copies of neighboring grid points in the local memory of each process (□). Hence, each process contains

*Parts of this work were funded by the GENESIS (ESPRIT # 2702) project

a so-called *overlap area* (surrounded by the dashed line in Figure 2) which, of course, has to be updated after each iteration step.

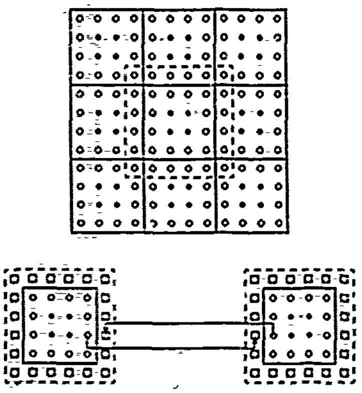


Figure 2: Overlap areas and their exchange.

The grid partitioning approach can be extended to block-structured grids in a straight-forward manner.

2.4 Multigrid methods

Standard iterative multigrid algorithms process a cycle from the fine to the coarse grids and back to the fine grids sequentially, whereas on each grid level the actual problem is treated in parallel similarly to the parallel single grid algorithms described in the previous sections.

On parallel distributed- memory systems an efficient implementation of multigrid algorithms is not trivial since the performance may degrade due to:

- idle processors on very coarse grids.
- short messages and dominant influence of start-up time.
- bad communication/computation ratio.

The algorithmical and technical details of parallel multigrid algorithms are described in [3].

2.5 Communications library

For grid applications, the explicit programming of the communication can be hidden from the user. In the SUPRENUM project, for example, a library of communication routines has been developed [1] which ensures

- clean and error-free programming,
- easy development of parallel codes.
- portability within the class of distributed memory computers. Programs can be ported to any of these machines as soon as the communication library has been implemented.

The library supports regular and block-structured grids and is available at the GMD or at the PALLAS GmbH.

3 Performance

3.1 Performance measures

The quantities of interest in evaluating the performance of parallel algorithms are:

- Time $T(N,P)$. time to solve a preblem of size N on a multiprocessor system using P nodes,
- speed-up $S(N,P) := T(N,1)/T(N,P)$,
- efficiency $E(N,P) := S(N,P)/P$.

Note that on the MIMD/SIMD architectures the utilization of the hardware capabilites is the product of the "multiprocessor" efficiency as defined above and the efficiency related to the vector processing unit. The total problem solving time - which is the only interesting number from the user's point of view - depends, of course, additionally on the *numerical efficiency* of an algorithm.

In practice E will be smaller than its ideal value 1, mainly because of communication (including synchronization), unbalanced load, and sequential parts in the algorithm.

3.2 Performance model for grid applications

Let N be the number of grid-points, D the dimension of the grid (typically $2 \leq D \leq 4$) and $n = N/P$ the size of the local grid on each processor The total grid is assumed to be cubic. Then the time for the arithmetic calculations is $T_{cal} = c_1 n$. On homogeneous parallel architecures the time needed for communication consits of two components, the so-called start-up time needed for the initialization of a message, and the transfer time which is proportinional to the lentgh of the message.

The data volume to be communicated depends on the way how the grid is partitioned. If the partitioning is performed in all D dimensions and cubic subgrids are generated the communication time is $T_{comm} = c_2 n^{(D-1)/D} + c_3$. The speed-up is

$$S(N,P) = \frac{P}{1 + \frac{c_2}{c_1} n^{(-1)/D} + \frac{c_3}{c_1} n^{-1}}.$$

The asymptotic behaviour is

$$S(N,P) \longrightarrow P,\ E(N,P) \longrightarrow 1 \qquad \text{for } n \longrightarrow \infty.$$

The efficiency remains constant for scaled problems (n constant, $P \longrightarrow \infty$).

The constants c_i depend on the particular grid algorithm and the properties of the hardware (communication speed, floating-point performance etc.). A detailed analysis can be found in [2].

References

[1] Hempel, R.: The SUPRENUM communications subroutine library for grid-oriented problems. Report ANL-87-23, Argonne National Laboratory, 1987.

[2] Solchenbach, K.: Performance evaluation for single and multi grid algorithms on multiprocessor systems with distributed memory. In [4].

[3] Solchenbach, K., Thole, C.A., Trottenberg, U.: Parallel multigrid methods: Implementation on SUPRENUM-like architectures and applications. In. Supercomputing. Proceedings of 1st International Conference on Supercomputing, June 8 12, 1987 in Athens. Lecture Notes in Computer Science 297, Springer Verlag, New York, 1988.

[4] Trottenberg, U. (ed.): Proceedings of the 2nd International SUPRENUM Colloquium "Supercomputing based on parallel computer architectures". Parallel Computing 7, North Holland, 1988.

IMACS session : Solution of P.D.E. on Massively Parallel MIMD Systems

P. LECA
ONERA, Parallel Computing Division
B.P. 72, 92322 Chatillon Cedex,
France

Abstract

The success of highly parallel distributed memory multiprocessors will depend mainly on their efficiency when running realistic application codes. This paper presents the main topics discussed in the session dedicated to the use of massively parallel MIMD systems in the field of scientific computing.

In the race to Teraflops performances a new generation of highly parallel multiprocessors is emerging, which is based on the use of a large set of powerfull microprocessors. Nevertheless the acceptance of such systems in the industry will depend mainly on their actual performance when running realistic application codes.
This could be achieved by redesigning numerical methods and algorithms that are used today to solve partial differential equations in areas such as CFD, structural analysis or electromagnetism simulation.

This session focuses on the development and the implementation of these methods on massively parallel MIMD architectures (iPSC, NCUBE, BBN ...).
On such systems the efficiency is often the result of a trade-off between the reduction of the time due to data communication, either in the communication network or through the memory hierarchy, and the reduction of the computation time. Then parallel algorithms exploiting data locality in private or local memory are specially stressed.
Moreover, the use of a geometric parallelism, based on a partition of the computational domain, pushes the development of software tools that provide automatically this partitioning and the corresponding data structures.
This subject has been studied at the University of Colorado where several mesh decomposers has been developped and adapted to various machines such as the hypercube iPSC2, the CRAY multiprocessors and the Connection Machine.
Furthermore representatives from the centers of NASA Ames and NASA ICASE present their last results about the utilization of massively parallel computers for CFD. The respective advantages of SIMD and MIMD computers are particularly discussed.
Then, the experience done at CERFACS concerning the development of a domain decomposition method is presented. This method, that provides a coarse grain parallelism, has been implemented with success on the iNTEL iPSC2.

The multigrid technique is now widely used for accelerating the convergence of iterative algorithms on structured grid. However this technique leads to non-local memory references that could enter in contradiction with an efficient implementation on a distributed memory architecture. The experience gathered at SUPRENUM on this subject is also adressed in this session.
At last, dealing with very large dense matrices issued from an integral equation formulation, recent algorithmns developments done at ONERA adequated to the numerical computation of Helmholtz equation are presented with implementation results on CRAY-2 and iPSC2 multiprocessors.

PARALLEL SYNERGY:
CAN A PARALLEL COMPUTER BE MORE EFFICIENT THAN THE SUM OF ITS PARTS?[1]

Selim G. Akl
IEEE Senior Member
Department of Computing and Information Science
Queen's University, Kingston, Ontario K7L 3N6
CANADA

Abstract The two most popular models of sequential and parallel computation lead, once defined precisely, to a computational paradox. Specifically, we show that for a wide family of problems the cost of a PRAM solution is smaller than that of its RAM counterpart This contradicts the currently established belief, and does not appear to be amenable to explanation using existing approaches. We use the term *parallel synergy* to refer to this phenomenon.

1. INTRODUCTION

Over the last forty years the Random Access Machine (RAM) has established itself as the most widely understood and used model of sequential computation [Engeler, Hopcroft, Machtey, Mandrioli]. The model consists primarily of a processor and a random access memory The processor possesses a constant number of local registers, and operates under the control of a sequential algorithm. Each step of such an algorithm consists of (up to) three phases:

(i) a READ phase, where the processor reads a datum from the random access memory and stores it in one of its local registers;

(ii) a COMPUTE phase, where the processor performs an elementary operation (such as comparison, addition, etc...) on data in its local registers;

(iii) a WRITE phase, where the processor writes into the random access memory the result of some computation.

Each of the phases is assumed to take constant time, leading to a constant execution time per step. It is important to emphasize that the model as described assumes that each of the operands and results of an elementary operation fits into a single memory location. Thus, in the terminology of [Aho], we are using the "uniform cost criterion" (not to be confused with the *cost of a computation* defined below).

In parallel computation, the Parallel Random Access Machine (PRAM) appears to be the preferred model among theoretical computer scientists [Akl 1, Gibbons, Karp, Parberry]. Here several processors share a common random-access memory. As in the RAM, each processor has a constant number of local registers. All processors simultaneously execute the instructions of a parallel algorithm. Each step of such an algorithm consists of (up to) three phases:

(i) a READ phase where processors read data from the shared memory and store them in their local registers;

(ii) a COMPUTE phase where processors perform elementary operations on local data;

(iii) a WRITE phase where processors write results to the shared memory.

Each of these phases requires constant time, again leading to a constant time per step of the algorithm. Note that some processors may not execute one or two phases of a given step. Also, during a READ or WRITE, every processor may gain access to a different memory location. However, the PRAM gives rise to a number of variants depending on whether two or more processors are allowed to gain access simultaneously to the same memory location (for reading or for writing). In this paper we shall be concerned solely with that variant of the PRAM which disallows such simultaneous access to the same memory location. This model is known in the literature as the Exclusive-Read Exclusive-Write (EREW) PRAM.

Let us define the *cost* of an algorithm as being the product of the *number of processors* it uses and its *running time* [Akl 1, Quinn]. (Note that some authors use the term *work* instead of cost [Cormen].) It is usually said that if c is the cost of running an algorithm A on a PRAM, then the cost of simulating A on a RAM is (asymptotically) equal to c [Akl 1, Almasi, Eager, Faber 1, Faber 2, Fishburn, Modi]. We argue in this paper that this statement is no longer true once the network interconnecting processors to memory locations is taken into consideration by the cost analysis. More precisely, we show that for a wide family of computational problems, the cost of a PRAM solution is smaller (asymptotically) than that of the best possible sequential solution. Furthermore, the cost of the PRAM solution grows when simulated on a RAM. We emphasize here that the question addressed in this paper (namely inclusion of the interconnection network in the cost analysis) is distinct from the problem treated in [Akl 3, Gnt, Gustafson, Janssen, Kornfeld, Lai, Leach, Li 1, Li 2, Mehrotra, Parkinson, Preiss, Quinn, Weide], where various issues pertaining to *speedup* (i.e. the ratio of sequential to parallel running time) are discussed.

2. TRADITIONAL COST ANALYSIS

Assume that a problem P of size n is given, for which A_s is a sequential algorithm running in sequential time $t_s(n)$ on a RAM. The cost of A_s is $c_s(n) = 1 \times t_s(n) = t_s(n)$. Note that, because there is only one processor, the cost of a sequential algorithm is exactly its running time.

Further, let A_p be a parallel algorithm for P running in parallel time $t_p(n) = t_s(n)/n$ on a PRAM with n processors. The cost of A_p is $c_p(n) = n \times t_p(n) = t_s(n)$.

As $c_s(n)$ and $c_p(n)$ are derived for the RAM and PRAM, respectively, they do not take into consideration the network required for memory access. Indeed, both the RAM and the PRAM ignore that network despite the fact that its cost (i.e. the product of the number of processors it uses and the time required to traverse it) dominates that of many computations. We now propose to examine what happens to $c_s(n)$ and $c_p(n)$ once the cost of the interconnection network is taken into account.

We note in passing that an interconnection network that does not allow *feedback*, i.e. a network where each processor is used once per memory access, is sometimes referred to as a *circuit* [Parberry]. Occasionally, the cost (or *size*) of a circuit is expressed simply as the number of processors it uses, without multiplying the latter by the time it takes to traverse the circuit (also known as the circuit's *depth*). In this paper, we prefer to adhere to the standard definition of cost (namely, number of processors × running time), in order to avoid restricting our discussion to circuits. As it turns out, inclusion of the memory access time in the analysis affects our results only marginally.

3. TRUE EFFICIENCY

In the RAM, in order to gain access to any of n memory locations, the processor issues a log n - bit address. A network of size O(n) decodes this address in O(log n) time This network is, for example, a binary tree of processors (i.e. a "circuit" in the terminology defined earlier) [Kuck, Tanenbaum]. It has a cost of O(n log n).

In the EREW PRAM, n processors can gain access to n memory locations simultaneously (one memory location per processor) in $O(\log^a n)$ time using a network of $O(n \log^b n)$ processors where a and b are two constants. Typical values of a and b are given in Table 1, along with references to the corresponding networks. The cost of the network for memory access is therefore $O(n \log^d n)$, where d = a + b.

Network	a	b
[Batcher]	2	2
[Stone]	2	0
[Ajtai]	1	1
[Leighton]	1	0

Table 1. Values of a and b for typical networks used in memory access. Note that, unlike the networks of [Stone] and [Leighton], those of [Batcher] and [Ajtai] are "circuits".

[1] This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant A-3336.

The revised cost of A_s is therefore,

$$c'_s(n) = (1 + O(n)) \times (t_s(n) \times O(\log n)) = t_s(n) \times O(n \log n),$$

while the revised cost of A_p is:

$$c'_p(n) = (n + O(n \log^b n)) \times (t_s(n)/n \times O(\log^a n)) = t_s(n) \times O(\log^d n).$$

In other words, the cost of the parallel algorithm is asymptotically smaller than that of the sequential algorithm! Notice that, because $t_p(n)$ is defined as $t_s(n)/n$, the exact value of $t_s(n)$ is irrelevant when computing the ratio $c'_s(n)/c'_p(n)$. If so needed, one may of course assume that $t_s(n)$ is the running time of the fastest possible algorithm for P.

It should also be clear that our analysis holds for any number of (traditional) PRAM processors, provided that this number is at most a linear function of n (the number of PRAM memory locations required to solve a given problem of size n). Indeed, denoting the number of PRAM processors by N, where N < n, we see that the asymptotic value of $c'_p(n)$ remains unchanged, namely,

$$\begin{aligned} c'_p(n) &= (N + O((n+N) \log^b (n+N)) \times (t_s(n)/n \times O(\log^a (n+N)) \\ &= t_s(n) \times O(\log^d n). \end{aligned}$$

As noted above, the traditional approach to analyzing parallel algorithms does not take into consideration the cost of the network interconnecting processors to memories. In that approach, the *efficiency* of a parallel algorithm for a given problem is defined as the ratio of the running time of the fastest sequential algorithm for that problem to the cost of the parallel algorithm. Because of the assumption that a RAM can simulate a PRAM algorithm in no more time than the cost of the latter, the efficiency of a parallel algorithm is at most 1. By contrast, we refer in this paper to the ratio $c'_s(n)/c'_p(n)$ as the *true efficiency* of a parallel algorithm. This ratio in the case of A_s and A_p is $O(n/\log^{d-1} n)$, which is larger than 1. In what follows, we assume that the circuit of [Ajtai] is used to implement the EREW PRAM, i.e. $d = 2$.

4. EXAMPLE

Consider the following problem. We are given n distinct integers $I_1, I_2, \ldots, I_n$ in the range $(-\infty, n]$, stored in an array $X[1], X[2], \ldots, X[n]$ in such a way that $X[i] = I_i$ for all $1 \le i \le n$. It is required to modify X so that it satisfies the following condition: for all $1 \le i \le n$, $X[I_i] = I_i$, if and only if $1 \le I_i \le n$, otherwise $X[i] = I_i$ [Akl 3].

Sequentially, the problem can be solved on the RAM in the obvious way in time $t_s(n) = O(n)$, and this is optimal since every entry of X must be examined once. Consequently, $c'_s(n) = O(n^2 \log n)$. In parallel, $t_p(n) = O(1)$ on a PRAM with n processors. Thus, $c'_p(n) = O(n \log^2 n)$. It follows that $c'_s(n)/c'_p(n) = O(n/\log n)$. Note that simulating the PRAM algorithm on the RAM leads to an algorithm whose cost is

$$O(n) \times (c'_p(n) \times \log n) = O(n^2 \log^3 n).$$

This cost is larger than the (optimal) costs of both the RAM and PRAM solutions.

5. GENERALIZATION

In general, for any computational problem of size n, $c'_s(n)/c'_p(n) > 1$, i.e. $(t_s(n) \times n \log n) / (t_p(n) \times n \log^2 n) > 1$, provided that $t_s(n)/t_p(n) > \log n$, and the number of steps where a READ and/or a WRITE phase is executed dominates the computation. This condition is satisfied by a wide family of problems. These problems include selection, merging, sorting, and a variety of computations in numerical analysis, graph theory and computational geometry [Akl 1].

6. DISCUSSION

Traditional analyses of cost either ignore the existence of a network to interconnect processors to memory locations, or (implicitly) assume that the cost of such a network is O(1). Both approaches are clearly unrealistic: any reasonable model of computation must include as an integral part a means of linking processors to memory locations, whose cost is a function of the number of these processors and memory locations.

On the other hand, in deriving $c'_s(n)$ and $c'_p(n)$ we have taken into account both the number of processors required for such a network and the time elapsed during memory access. It may be argued that since the processors used to build the interconnection network are simpler than those actually doing the arithmetic and logical computations, the two ought not be treated equally in the cost analysis. The fallacy in this argument is that the complexity (i.e. size and number of internal components such as registers) of a (RAM or PRAM) processor used for computing is only a constant multiple of that of an interconnection network processor (since both are expected to handle data of the same magnitude). It is therefore quite reasonable in an asymptotic cost analysis to lump the two kinds of processors together and view them (as we did) as *active agents* of a computational model (memory locations being the *passive agents*). As a result, we arrived at a conclusion contradicting the established belief whereby the cost of a parallel algorithm for a given problem cannot be smaller than that of the best possible sequential algorithm for that problem.

In an attempt to resolve this paradox, we may use the following compromise: we (explicitly) assume that the interconnection network is part of the RAM and PRAM, but that its cost in both models is O(1). However, this solution leads, in turn, to a result not unlike the one reached in the previous section. Using an approach developed in [Akl 2], [Akl 5], and [Akl 6], we show in [Akl 4] that a model significantly more powerful than the PRAM can be obtained by extending the latter to include a network whose cost is asymptotically equal to that required to interconnect processors and memory locations in the EREW PRAM. Assuming that the cost of that network (as in the PRAM) is O(1), we obtain solutions to a variety of problems whose cost is smaller than that of the best known PRAM solutions.

7. CONCLUSION

From the above discussion we conclude that the PRAM allows for a synergistic phenomenon to occur. This phenomenon manifests itself by a reduction in computational cost (as defined in this paper) for a wide variety of problems. These problems are characterized by an intensive movement of data from and into memory during the course of a computation. On the RAM, each access to *one* of n memory locations requires O(log n) time, and uses an interconnection network of size O(n). The PRAM, on the other hand, allows access to *all* n memory locations simultaneously (also in O(log n) time) via an interconnection network of size O(n log n), *and not* $O(n^2)$. Through what we call *parallel synergy*, a PRAM with n processors is therefore more efficient than n RAMs.

This result has a number of implications. Our work was originally motivated by the observation that both the RAM and the PRAM, *as theoretical models of computation*, are severely lacking. Indeed, neither model takes into account the cost of such a fundamental operation as memory access. By defining the abstract model more precisely to include all important operations, not only do we get a more realistic and meaningful analysis, but we also uncover hitherto unknown phenomena. Finally, as noted above, both the RAM and the PRAM are idealized computers. From the practical point of view, it may be useful to conduct analyses of true efficiency, as defined in this paper, for real computers.

8. REFERENCES

[Aho]

Aho, A.V., Hopcroft, J.E., and Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Massachusetts, 1974.

[Ajtai]

Ajtai, M., Komlos, J., and Szemeredi, E., An O(n log n) sorting network, Proceedings of the 15th Annual Symposium on Theory of Computing, Boston, Massachusetts, May 1983, pp. 1 - 9.

[Akl 1]

Akl, S.G., The Design and Analysis of Parallel Algorithms, Prentice- Hall, Englewood Cliffs, New Jersey, 1989.

[Akl 2]

Akl, S.G., On the power of concurrent memory access, in: Computing and Information, North-Holland, Amsterdam, 1989, pp. 49 - 55.

[Akl 3]
Akl, S.G., Cosnard, M., and Ferreira, A.G., Data-movement-intensive problems: Two folk theorems in parallel computation revisited, Technical Report No. 90-18, Laboratoire de l'Informatique du Parallelisme, Ecole Normale Superieure de Lyon, Lyon, France, June 1990.

[Akl 4]
Akl, S.G., and Fava, L., An efficient interconnection network for BSR, manuscript in preparation, July 1990.

[Akl 5]
Akl, S.G., and Guenther, G.R., Broadcasting with selective reduction, Proceedings of the 11th IFIP Congress, San Francisco, California, August 1989, pp. 515 - 520.

[Akl 6]
Akl, S.G., and Guenther, G.R., Reflections on a parallel model of computation, submitted for publication.

[Almasi]
Almasi, G.S., and Gottlieb, A., Highly Parallel Computing, Benjamin/ Cummings, Redwood City, California, 1989.

[Batcher]
Batcher, K.E., Sorting networks and their applications, Proceedings of the AFIPS 1968 Spring Joint Computer Conference, Atlantic City, New Jersey, April 1968, pp. 307 - 314.

[Cormen]
Cormen, T.H., Leiserson, C.E., and Rivest, R.L., Introduction to Algorithms, The MIT Press, Cambridge, Massachusetts, 1990.

[Eager]
Eager, D.L., Zahorjan, J., and Lazowska, E.D., Speedup versus efficiency in parallel systems, IEEE Transactions on Computers, Vol. C-38, No. 3, 1989, pp. 408 - 423.

[Engeler]
Engeler, E., Introduction to the Theory of Computation, Academic Press, New York, 1973.

[Faber 1]
Faber, V., Lubeck, O.M., and White, A.B. Jr., Superlinear speedup of an efficient sequential algorithm is not possible, Parallel Computing, Vol. 3, 1986, pp. 259 - 260.

[Faber 2]
Faber, V., Lubeck, O.M., and White, A.B. Jr., Comments on the paper: "Parallel efficiency can be greater than unity", Parallel Computing, Vol. 4, 1987, pp. 209 - 210.

[Fishburn]
Fishburn, J.B., Analysis of Speedup in Distributed Algorithms, UMI Research Press, Ann Arbor, Michigan, 1981.

[Gibbons]
Gibbons, A., and Rytter, W., Efficient Parallel Algorithms, Cambridge University Press, Cambridge, England, 1988.

[Grit]
Grit, D.H., and McGraw, J.R., Programming divide and conquer for a MIMD machine, Software Practice and Experience, Vol. 15, No. 1, 1985, pp. 41 - 53.

[Gustafson]
Gustafson, J.L., Revaluating Amdahl's law, Communications of the ACM Vol. 31, No. 5, 1988, pp. 532 - 533.

[Hopcroft]
Hopcroft, J.E., and Ullman, J.D., Introduction to Automata, Languages, and Computation, Addison-Wesley, Reading, Massachusetts, 1979.

[Janssen]
Janssen, R., A note on superlinear speedup, Parallel Computing, Vol. 4, 1987, pp. 211 - 213.

[Karp]
Karp, R.M., and Ramachandran, V., A survey of parallel algorithms for shared memory machines, in: Handbook of Theoretical Computer Science, North- Holland, Amsterdam, 1989.

[Kornfeld]
Kornfeld, W.A., Combinatorially implosive algorithms, Communications of the ACM, Vol. 25, No. 10, 1982, pp. 734 - 738.

[Kuck]
Kuck, D.J., The Structure of Computers and Computations, Vol. 1, John Wiley & Sons, New York, 1978.

[Lai]
Lai, T., and Sahni, S., Anomalies in parallel branch and bound algorithms, Communications of the ACM, Vol. 27, No. 6, 1984, pp. 594 602.

[Leach]
Leach, R.J., Atogi, M., and Stephen, R.P., The actual complexity of parallel evaluation of low degree polynomials, Parallel Computing, Vol. 13, 1990, pp. 73 - 83.

[Li 1]
Li, G.J., and Wah, B.W., Coping with anomalies in parallel branch and bound algorithms, IEEE Transactions on Computers, Vol. C 35, No 6, 1986, pp. 568 - 573.

[Li 2]
Li, K., IVY: A shared virtual memory system for parallel computing, Proceedings of the International Conference on Parallel Processing, St. Charles, Illinois, August 1988, pp. 94 - 101.

[Leighton]
Leighton, F.T., Tight bounds on the complexity of parallel sorting, IEEE Transactions on Computers, Vol. C-34, No. 4, 1985, pp. 344 - 354.

[Machtey]
Machtey, M., and Young, P., An Introduction to the General Theory of Algorithms, North-Holland, New York, 1978.

[Mandrioli]
Mandrioli, D., and Ghezzi, C., Theoretical Foundations of Computer Science, John Wiley & Sons, New York, 1987.

[Mehrotra]
Mehrotra, R., and Gehringer, E.F., Superlinear speedup through randomized algorithms, Proceedings of the International Conference on Parallel Processing, St. Charles, Illinois, August 1985, pp. 291 - 300.

[Modi]
Modi, J.J., Parallel Algorithms and Matrix Computation, Clarendon Press, Oxford, England, 1988.

[Parberry]
Parberry, I., Parallel Complexity Theory, John Wiley & Sons, New York, 1987.

[Parkinson]
Parkinson, D., Parallel efficiency can be greater than unity, Parallel Computing, Vol. 3, 1986, pp. 261 - 262.

[Preiss]
Preiss, B.R., and Hamacher, V.C., Semi-static dataflow, Proceedings of the International Conference on Parallel Processing, St. Charles, Illinois, August 1988, pp. 127 - 134.

[Quinn]
Quinn, M.J., Designing Efficient Algorithms for Parallel Computers, McGraw-Hill, New York, 1987.

[Stone]
Stone, H.S., Parallel processing with the perfect shuffle, IEEE Transactions on Computers, Vol. C-20, No. 2, 1971, pp. 153 - 161.

[Tanenbaum]
Tanenbaum, A.S., Structured Computer Organization, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[Weide]
Weide, B.W., Modeling unusual behavior of parallel algorithms, IEEE Transactions on Computers, Vol. C-31, No. 11, 1982, pp. 1126 - 1130.

RELATIONAL STRUCTURE SEMANTICS OF CONCURRENT SYSTEMS

Ryszard Janicki
Department of Computer Science and Systems
McMaster University
Hamilton, Ontario, Canada, L8S 4K1
email: janicki@ca.mcmaster.dcss.maccs

Maciej Koutny
Computing Laboratory
The University of Newcastle upon Tyne
Newcastle upon Tyne NE1 7RU, U.K.
email: Maciej.Koutny@uk.ac.newcastle

Abstract We introduce a new relational structure semantics of concurrent systems which is a generalisation of the causal partial order semantics. The new semantics is consistent with the operational behaviour of inhibitor and priority nets expressed in terms of step sequences. It employs combined partial orders - *composets* (each composet is a *relational structure* consisting of a causal partial order and a weak causal partial order). We outline the way in which composets can be generated for 1-safe inhibitor nets.

1. INTRODUCTION

In the development of mathematical models for concurrent systems, the concepts of partial and total order undoubtedly occupy a central position. Interleaving models use total orders of event occurrences, while so-called 'true concurrency' models use step-sequences or causal partial orders (comp. [BD87;Ho85,Pr86]). Even more complex structures, such as failures [Ho85] or event structures [Wi32], are in principle based on the concept of a total or partial order. While interleavings and step sequences usually represent executions or observations and can be regarded as directly representing operational behaviour of a concurrent system, the causality relation represents a set of executions or observations. The lack of ordering between two event occurrences in the case of a step sequence is interpreted as simultaneity, while in the case of a causality relation it is interpreted as independency, which means that the event occurrences can be executed (observed) in either order or simultaneously. In other words, a causal partial order is an invariant describing an abstract history of a concurrent system. Both interleaving and true concurrency models have been developed to a high degree of sophistication and proved to be successful specification, verification and property proving tools. However, there are some problems, for instance the specification of priorities using partial orders alone is rather problematic (see [La85, Ja87, JL88]). Another example are inhibitor nets (see [Pe81]) which are virtually admired by practitioners, and almost completely rejected by theoreticians, in our opinion mainly because their concurrent behaviour cannot be properly defined in terms of causality based structures. We think that these kind of problems follows from the general assumption that all behavioural properties of a concurrent system can be adequately modelled in terms of causal partial orders or causality-based relations. We claim that the structure of concurrency phenomena is richer and there are other invariants which can represent an abstract history of a concurrent system. In this paper we will show how one of such invariants (*weak causal partial order*) can be defined and derived. Our main result will be the definition of a *relational structure* semantics for inhibitor nets which employs combined partial orders *composets* to model the interrelationship between event occurrences involved in a concurrent history (each composet is a *relational structure* consisting of a causal partial order and a weak causal partial order). The resulting semantics model is consistent with the operational behaviour of such nets expressed in terms of step sequences. We also show that the way in which composets can be generated is a direct generalisation of the procedure used to generate causal partial orders. The results of this paper are directly applicable to systems with priorities, nets with inhibitor arcs, bounded nets, and virtually any system model which supports the notion of true concurrency semantics.

2. MOTIVATION

In this section we present an example which we believe clearly identifies an inability of the causal partial order (CPO) semantics to cope properly with some important aspects of non-sequential behaviour. We will use Petri nets [Pe81,Re85] to illustrate our discussion.

Our example closely follows the discussion in [Ja87,JL88]. We consider a concurrent system *Con* comprising two sequential subsystems *A* and *B* such that:

(1) *A* can execute event *a* and after event *b*.
(2) *B* can either engage in event *b*, or engage in event *c*.
(3) The two sequential subsystems synchronise by means of the handshake communication.
(4) The specification of *Con* includes a priority constraint stating that whenever it is possible to execute event *b*, then event *c* must not be executed.

One can model *Con* as the Petri net N_{prior} in Figure 1. Before analysing the behaviour of N_{prior}, we look at the behaviour of net *N*, where *N* is N_{prior} without the priority con-

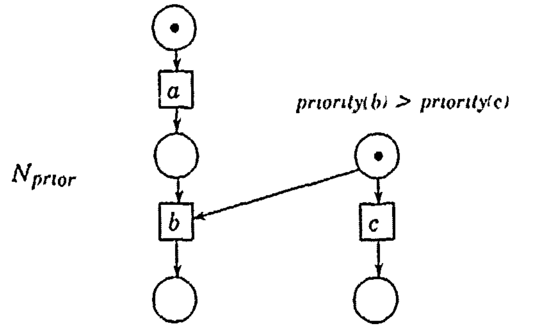


Figure 1

straint. The operational behaviour of N can be captured by the set of step sequences it generates, each step being a set of events executed simultaneously, as follows:

$steps(N)=\{\lambda, \{a\}, \{c\}, \{a\}\{b\}, \{a\}\{c\}, \{a,c\}, \{c\}\{a\}\}$.

Note that λ denotes the empty step sequence. A fundamental result of the theory of causal partial orders now says that there is a set of partially ordered sets, $posets(N)$, such that

$$steps(N)=\bigcup_{po\in posets(N)} steps(po) \tag{2.1}$$

where $steps(po)$ is the set of all step sequences consistent with a poset po. Figure 2 shows the elements of $posets(N)$ together with the sets $steps(po)$. Note that each $po\in posets(N)$ is interpreted as a causality relationship (an invariant) involving event occurrences, and is intended to represent an abstract history of net N. The consistency between the operational and invariant semantics captured by (2.1) is a cornerstone of the theory of causal partial orders.

Having looked at the two-level (i.e. operational and invariant) description of the behaviour of N, one might attempt to repeat the same construction for the priority net N_{prior}. It is relatively easy to obtain the operational semantics of N_{prior}. We simply delete form $steps(N)$ those step sequences which are inconsistent with the priority constraint. As the result we obtain:

$steps(N_{prior})=\{\lambda, \{a\}, \{c\}, \{a\}\{b\}, \{a,c\}, \{c\}\{a\}\}$.

Note that we deleted $\{a\}\{c\}$ since after executing a, event b becomes enabled and thus c cannot be executed. We should now be able to find a set $posets(N_{prior})$ such that

$$steps(N_{prior})=\bigcup_{po\in posets(N_{prior})} steps(po). \tag{2.2}$$

However, any attempt at finding such a set has to fail.

Proposition 1

There exists no set of partially ordered sets $posets(N_{prior})$ such that (2.2) holds. □

This leads us to a conclusion that it is impossible to construct a CPO semantics of the priority net N_{prior} which would be consistent with the full operational behaviour of that net, $steps(N_{prior})$. In order to develop an invariant semantics for N_{prior} which would be consistent with $steps(N_{prior})$, we must go beyond the CPO-based framework. In the rest of this paper we will show that there is a relational structure semantics of N_{prior}, called *combined partial order* semantics - $composets(N_{prior})$ - such that

$$steps(N_{prior})=\bigcup_{co\in composets(N_{prior})} steps(co). \tag{2.3}$$

We will show that the composet semantics is a generalisation of the CPO semantics, and that the way it may be derived is very similar to that used to derive the standard CPO semantics. We finally note that in this paper by a partially ordered set (poset) we mean a pair (X,R) such that X is a set, and $R\subseteq X\times X$ is irreflexive ($\neg aRa$) and transitive ($aRb \wedge bRc \Rightarrow aRc$).

$po\in posets(N)$	$steps(po)$
$\varnothing$	λ
a •	$\{a\}$
c •	$\{c\}$
a •⟶• b	$\{a\}\{b\}$
a • • c	$\{a\}\{c\}, \{a,c\}, \{c\}\{a\}$

Figure 2

3. AN OUTLINE OF THE COMPOSET MODEL

In this and the following sections by a step sequence we will mean a special kind of labelled partial order (for which the un-ordering relation is transitive) rather than a sequence of sets of events. The two representations are isomorphic, and the translation from the latter to the former is illustrated in Figure 3 (see [Ja87] for details).

If we look closer at the CPO model, it turns out that an abstract history H of a concurrent system can be represented in either of the following two forms:

(3.1) On the invariant level - as a poset which captures the causality relationship between the event occurrences involved in history H.

(3.2) On the operational level - as a set of step sequences being underlain by the same causal relationship, i.e., as $steps(po)$ for some po from (3.1).

In addition, we have some properties which establish the consistency between these two different views on H.

(3.3) Each poset po can be realised on the operational level, i.e., $steps(po)\neq\varnothing$.

(3.4) Each poset is uniqely identified by the step sequences which are consistent with it, i.e., $steps(po)=steps(po')$ implies $po=po'$.

(3.5) Each poset po is indeed an invariant, i.e., a precedes b in po if and only if a precedes b in every step sequence in $steps(po)$.

The main reason why posets are adequate representations of concurrent histories in the CPO model is that the whole approach is founded upon the following assumption concerning the relative order of two event occurrences, a and b, involved in a concurrent history H.

(3.6) The existence of a step sequence in H in which a and b occur simultaneously *is equivalent* to the existence of two step sequences in H, one in which a precedes b, the other in which a follows b.

From (3.6) it follows that a poset can be an adequate representation of a concurrent history H. There are exactly three possible invariant relationships between two event occurrences, a and b, involved in H.

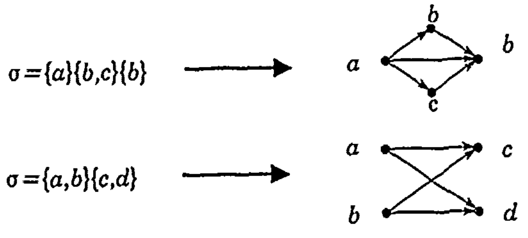


Figure 3

(3.7) In all step sequences in H, a precedes b.

(3.8) In all step sequences in H, a follows b.

(3.9) There are $\sigma_1,\sigma_2,\sigma_3$ in H such that: a precedes b in σ_1; a follows b in σ_2; and a is simultaneous with b in σ_3.

Then (3.7) and (3.8) can be represented on the invariant level by having a and b ordered in an appropriate way (a and b are causally dependent), while (3.9) can be captured by having no order between a and b (a and b are independent).

As we have already seen, N_{prior} does not satisfy (3.6), since $\{a,c\} \in steps(N_{prior})$ while $\{a\}\{c\} \notin steps(N_{prior})$. More precisely, the left-to-right part of (3.6) may fail to hold in systems like N_{prior}, while the right-to-left implication still holds. In [JK90] and [JK91] it has been shown that this implies that on the invariant level the partial orders have to be replaced by more complex relational structures, which we will call *combined partial orders* (or *composets*). A composet is a *relational structure* (see [Co81]) $co=(\Sigma,\rightarrow,\nearrow)$ such that $(\Sigma,\rightarrow)$ is the standard causality invariant, and $(\Sigma,\nearrow)$ is a *weak causality* invariant. The weak causality essentially means that if $a \nearrow b$ then in all step sequences consistent with co, *a precedes or is simultaneous with b*. In this way it is possible to capture three additional invariant relationships between two event occurrences involved in a concurrent history:

(3.10) In all σ in H, a is simultaneous with b.

(3.11) There are step sequences in H, σ_1 and σ_2, such that: a precedes b in σ_1; a is simultaneous with b in σ_2; and there is no step sequence in H in which a follows b.

(3.12) There are step sequences in H, σ_1 and σ_2, such that: a follows b in σ_1, a is simultaneous with b in σ_2; and there is no step sequence in H in which a precedes b.

Indeed, (3.11) and (3.12) can be represented on the invariant level by saying that a and b are weakly ordered in an appropriate way but not causally ordered, and (3.10) can be captured by having both weak causal orders between a and b (a and b are *synchronised*).

Having extended the posets to composets one can define the set of step sequences consistent with a composet co, $steps(co)$, and prove that (2.3) holds for a suitably defined set of composets of N_{prior} (see the Section 5). In this way the CPO approach which has proven to be so fruitful for systems satisfying (3.6) can be extended to concurrent systems for which (3.6) may not hold. For a detailed discussion and, in particular, the proof that composet is an adequate notion of history invariant for systems which may not satisfy the left-to-right implication in (3.6) but satisfy the right-to-left implication, the reader is advised to refer to [JK90] and [JK91].

4. THE MODEL

We define a *composet* (or *combined partially ordered* set) as a relational structure (see [Co81]) $co=(\Sigma,\rightarrow,\nearrow)$ such that Σ is a finite set of event occurrences and $\rightarrow$, $\nearrow$ are relations on Σ satisfying the following.

1. $\neg a \nearrow a$
2. $a \nearrow b \Rightarrow \neg b \rightarrow a$
3. $a \rightarrow b \Rightarrow a \nearrow b$
4. $a \nearrow b \wedge b \nearrow c \Rightarrow a=c \vee a \nearrow c$
5. $a \nearrow b \wedge b \rightarrow c \Rightarrow a \rightarrow c$
6. $a \rightarrow b \wedge b \nearrow c \Rightarrow a \rightarrow c$.

Corollary 2

$(\Sigma,\rightarrow)$ is a partially ordered set, while $(\Sigma,\nearrow)$ is a pre-ordered set. □

The two relations constituting a composet can be derived as invariants of a set of step sequences with the same domain.

Proposition 3

Let Δ be a non-empty set of step sequences with a common domain Σ. Let $\rightarrow_\Delta$ and $\nearrow_\Delta$ be two relations on Σ defined by

$a \rightarrow_\Delta b \quad :\Leftrightarrow \forall \sigma \in \Delta.\ a \rightarrow_\sigma b$

and $a \nearrow_\Delta b \quad :\Leftrightarrow \forall \sigma \in \Delta.\ a \rightarrow_\sigma b \vee a \leftrightarrow_\sigma b$

where $a \rightarrow_\sigma b$ and $a \leftrightarrow_\sigma b$ respectively means that a precedes b and a is simultaneous with b in σ.

Then $co=(\Sigma,\rightarrow_\Delta,\nearrow_\Delta)$ is a compose.. □

I.e., a composet can in a natural way be derived as an invariant of a set of step sequences. To show that a composet is an adequate invariant representation we need another result.

Let $co=(\Sigma,\rightarrow,\nearrow)$ be a composet. The set of step sequences consistent with co, $steps(co)$, comprises all step sequences σ with the domain Σ satisfying the following.

1. $a \rightarrow b \quad \Rightarrow \quad a \rightarrow_\sigma b$
2. $a \nearrow b \quad \Rightarrow \quad a \rightarrow_\sigma b \vee a \leftrightarrow_\sigma b$.

Then the adequacy of the composet notion follows from the following result (comp. (3.5)).

Theorem 4

If co is a composet and $\Delta = steps(co)$ then $co = (\Sigma,\rightarrow_\Delta,\nearrow_\Delta)$ □

The next result is a direct generalisation of the properties of the CPO model captured by (3.3) and (3.4).

Theorem 5

1. For every composet co, $steps(co) \neq \varnothing$.
2. If $steps(co)=steps(co')$ then $co=co'$. □

The composets together with the *steps* operation can provide an invariant model in exactly the same way as the causal partial orders. To show that the new model overcomes the shortcomings of the CPO model, we now have the following.

Proposition 6

There is a set of composets, $composets(N_{prior})$, such that (2.3) holds. □

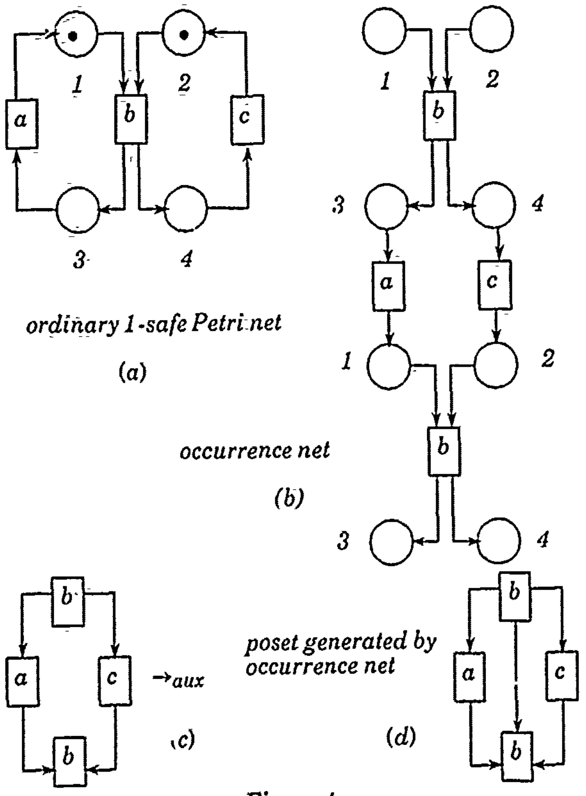
In this way we have solved the problem from Section 2, i e., we have found an invariant semantics of N_{prior} which in a direct way generalises the CPO semantics (note that each poset $(\Sigma,\rightarrow)$ is isomorphic to the composet $(\Sigma,\rightarrow,\rightarrow)$), and is consistent with the full operational semantics of N_{prior}. Note that the abstract histories of N_{prior} may be represented either as composets or as sets of step sequences which are consistent with those composets.

There is a certain similarity between our definition of the composet and the axioms for strong and weak precedence relation presented in [La86]. However, the way these two concepts are derived, the motivations, and the reasons for their introduction are quite different. Hence this similarity is either accidental or, as we would suggest, the composet is a natural generalisation of the concept of the partial order, and it may be useful for various, perhaps unrelated, applications.

5. THE CONSTRUCTION OF COMPOSETS

In this section we outline an algorithm for constructing the set of composets of a concurrent system. Since in our construction we can use a number of notions which have been developed for ordinary Petri nets, we will show the construction for 1-safe inhibitor nets [Pe81]. Note that 1-safeness means that each place may hold at most one token, and an inhibitor net is a Petri net with added inhibitor arcs. An inhibitor arc between place *p* and transition (event) *t* means that *t* can be enabled only if *p* is not marked. In the diagrams an inhibitor arc is identified by a small circle at one end.

The standard approach in which the CPO semantics for ordinary 1-safe Petri nets is derived employs *occurrence nets* (see [Re85, BD87]). An occurrence net is a representation of the causality relation on event occurrences (or single abstract history of the net). It is an unmarked acyclic net whose each place has at most one input and one output transition. Occurrence nets can be obtained by unfolding marked nets and resolving the conflicts according to the the firing rules. This is illustrated in Figure 4(a,b). Each occurrence net induces a poset on event occurrences in the following way: First an auxiliary relation $\rightarrow_{aux}$ is derived by transforming each three-node path *event1*$\rightarrow$*place*$\rightarrow$*event2* in the graph of the occurrence net into a pair *event1*$\rightarrow_{aux}$*event2*. Then a CPO is obtained by taking the transitive closure of $\rightarrow_{aux}$. For the occurrence net of Figure 4(b), the auxiliary relation $\rightarrow_{aux}$ is



ordinary 1-safe Petri net

(a)

occurrence net

(b)

(c)

poset generated by occurrence net

(d)

Figure 4

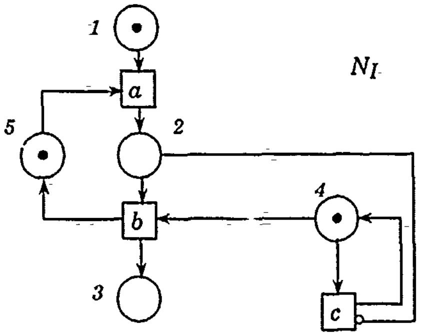


Figure 5

shown in Figure 4(c), and the resulting causal partial order is shown in Figure 4(d).

The way in which we construct composets for an inhibitor net will closely follow the above procedure. Let N_I be the inhibitor net shown in Figure 5. We first define an *occurrence net* of an inhibitor net by generalising in a straightforward way the standard definition of an occurrence net of an ordinary Petri net. The only new element is the handling of the inhibitor arcs. Since in the occurrence net places represent tokens, it is not possible to join *c* with place *2* using an inhibitor arc. However, we can join *c* with the *complement place* [Re85] of *2*, i.e. place *5*, using an activator arc (with a black dot at one end). Intuitively, this means that c can be executed only when *5* is marked. We also note that there is no restriction on the number of activator arcs which can be adjacent to a single place. A possible occurrence net for the inhibitor net N_I is shown in Figure 6.

The next step is to transform the structural relationships embedded in the graph of the occurrence net into two auxiliary relations, $\rightarrow_{aux}$ and $\nearrow_{aux}$, from which the composet can be derived. There are three structural relationships between event occurrences which we need to consider, as shown in Figure 7. For the occurrence net of Figure 6 the two auxiliary relations are shown in Figure 8(a).The final step has to take into account the various transitivities from the definition of a composet. More precisely, if $\rightarrow_{aux}$ and $\nearrow_{aux}$ have been defined for an occurrence net *ON* with Σ being the set of event occurrences, then the composet induced by *ON*, is defined as $co(ON)=(\Sigma,\rightarrow,\nearrow)$, where $(\Sigma,\rightarrow,\nearrow)$ is a minimal composet (w.r.t. set inclusion for both $\rightarrow$ and $\nearrow$) such that

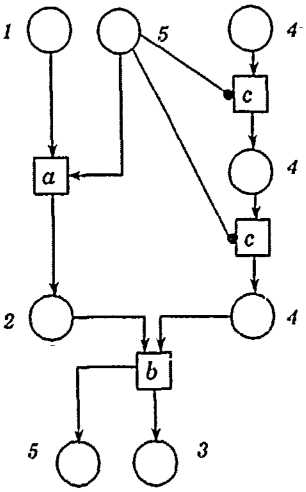


Figure 6

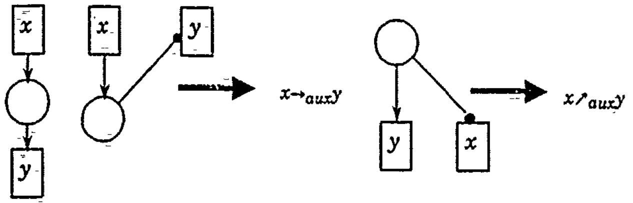


Figure 7

$a \rightarrow_{aux} b \Rightarrow a \rightarrow b$ and $a \nearrow_{aux} b \Rightarrow a \nearrow b$. For the occurrence net of Figure 6, the resulting → and ↗ are shown in Figure 8(b). We also note that in this case

$$steps(co(ON)) = \{ \{c\}\{a,c\}\{b\},\ \{c\}\{c\}\{a\}\{b\} \}.$$

For every inhibitor net IN, let $ON(IN)$ be the set of its occurrence nets. We define the invariant semantics of IN as follows: $composets(IN) = \{co(ON) \mid ON \in ON(IN)\}$.

Theorem 7

$steps(IN) = \bigcup_{co \in composets(IN)} steps(co)$. □

I.e., there exists an invariant semantics for inhibitor nets which is consistent with their operational semantics. We strongly emphasise that it is <u>not</u> possible to define a CPO semantics for the inhibitor net in Figure 5, which would satisfy the same consistency criterion. The reason is that we have $\{a,c\} \in steps(N_I)$ while $\{a\}\{c\} \notin steps(N_I)$. The construction of the relational structure semantics for inhibitor nets outlined in this section gives indirectly the relational structure semantics for N_{prior}, since each 1-safe priority net can be transformed into an operationally equivalent inhibitor net. For N_{prior} an equivalent inhibitor net can be obtained from that in Figure 5 by deleting the arc joining c and 4. The construction described in this section can be extended to other kinds of nets and priority models (see [JL88]). A bridge joining the net-based model with other models for concurrency could be provided by [Ta89].

Final Comments

We believe that in order to cope properly with general concurrent behaviours one should not restrict ones concerns only to structures based on causal partial orders. The composets (i.e. causality and weak-causality) can provide an invariant semantics for priority systems and inhibitor nets. Although both priority nets and inhibitor nets have the power of Turing machines as far as their interleaving semantics is concerned ([Pe81]), one may show that their composet semantics are different (there are composets which can be generated by inhibitor nets but not by priority nets). By combining the above approach with the results of [Ja87, JL88] we can obtain a model successfully dealing with priority systems, as for instance the full *occam* programming language (see example [Ro84]). For more details concerning the approach presented here, the reader is advised to refer to [JK90] and [JK91].

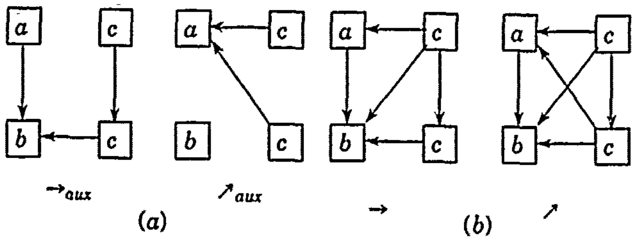


Figure 8

Acknowledgements

The research of the first author was partly supported by a grant from NSERC No. OGP 0036539. The research of the second author supported by ESPRIT Basic Research Action 3148 (project DEMON).

References

[BD87] Best E., Devillers R., *Sequential and Concurrent Behaviour in Petri Net Theory*, Theoretical Computer Science, 55 (1987), pp. 87-136.

[Co81] Cohn P.M., *Universal Algebra*, D. Reidel, 1981.

[Ho85] Hoare C.A.R., *Communicating Sequential Processes*, Prentice-Hall, 1985.

[Ja87] Janicki R., *A Formal Semantics for Concurrent Systems with a Priority Relation*, Acta Informatica 24, 1987, pp.33-55.

[JK90] Janicki R., Koutny M., *A Bottom Top Approach to Concurrency Theory Part I. Observations, Invariants and Paradigms*, Technical Report 90-04, McMaster University, 1990.

[JK91] Janicki R., Koutny M., *Invariants and Paradigms of Concurrency Theory*, Proc. of Parle'91, Lecture Notes in Computer Science, to appear, 1991.

[JL88] Janicki R., Lauer P.E., *On the Semantics of Priority Systems*, 17th Annual International Conference on Parallel Processing, Vol. 2, pp. 150-156, 1988, Pen. State Press.

[La85] Lamport L., *What It Means for a Concurrent Program to Satisfy a Specification: Why No One Has Specified Priority*, 12th ACM Symposium on Principles of Programming Languages, New Orleans, Louisiana, 1985, pp. 78-83.

[La86] Lamport L., *On Interprocess Communication, Part I. Basic formalism, Part II, Algorithms*, Distributed Computing 1 (1986), pp. 77-101.

[Pe81] Peterson J.L., *Petri Net Theory and the Modeling of Systems*, Prentice Hall, 1981.

[Pr86] Pratt V., *Modelling Concurrency with Partial Orders*, Int. Journal of Parallel Programming 15, 1 (1986), pp. 33-71.

[Re85] Reisig W., *Petri Nets*, Springer 1985.

[Ro84] Roscoe A.W., *Denotational Semantics for OCCAM*, Lecture Notes in Computer Science 197, Springer 1984, pp. 306-329.

[Ta89] Taubner D., *Finite Representations of CCS and TCSP Programs by Automata and Petri Nets*, Lecture Notes in Computer Science 369, Springer 1989.

[Wi82] Winskel G., *Event Structure Semantics for CCS and Related Language*, Lecture Notes in Computer Science 140, Springer 1982, pp. 561-567.

MINIMAL STATE CELLULAR SEGMENT GENERATION.

PAWEŁ P. SIWAK
Computer Science Centre
Technical University of Poznań
60-965 Poznań, Poland

Abstract. A 5-state solution of the problem of generating the stable segment of length L with 1-D scope-3 cellular automaton is presented. L is represented by the initial configuration in binary form of some polynomial $R(L)$. Here $L > 0$ and simply $R(L) = L$. Carry-free counter with states in RNS form and two waves transmitting the information were used in the solution. The generation needs $T = 2L + 3$ steps of time. Further, it is shown how $R(L)$ is related to the speed of the front wave. A comparison of some solutions is also given.

I. INTRODUCTION.

We focuse here on generating a stable 1-D final pattern, called the segment and composed of L consecutive states x starting from an initial configuration which contains certain encoded information about L. The problem is to design a minimal state 1D scope-3 cellular automaton (CA) capable to realize such transformation. This will be reffered to as CSG (cellular segment generation) problem. The problem was posed and partially solved in 1975 [1]. The immediate application of the CSG arise in VLSI circuits when L is assumed in its binary form; then the CSG actually performs highly regular parallel conversion of binary number L to its unary form.

An effort was undertaken on looking for the minimal state space in parallel production systems, especially in 2-D and 1-D CAs. In 2-D the 29-state scope-5 CA was first used by von Neumann. Soon, a system capable of universal computation has been improved and a 4-state solution has been given [5]. Also, Conway's 2-state scope-9 cellular "game of life" was shown to be universal. Recently, a 3-state scope-5 universal CA was proposed [6]. In 1-D the 'simple' computation-universal CAs have been recognized quite early. An 18-state scope-3 CA was proved [7] to belong to that class. In order to implement practically the embedding of computations in CAs some other problems are under research, too. The techniques intented to synchronize various events in 1-D CAs were mastered, as well [4]: an 8-state scope-3 and a 17-state scope-3 CAs for a firing squad problems were proposed [4]. Also a medium for performing computations, namely 1-D 2-state scope-$(2r+1)$ filter automata, a modification of CA model, was proposed [8], and even its VLSI implementation given.

The CAs capable of generating certain particular patterns are extensively searched for. Many configurations called "primitive elements", "basic organs" or "general-purpose components" were successfuly created to support simple computations. This effort is mainly motivated by expected VLSI implementations and applications; in computer graphics, code number conversion, random numbers [3] and possibly in VLSI layout generation.

There are two solutions [2] for the CSG problem since 1985. One, with 7-state cell, generates segments of length $L > 1$ in time $T = 3L + r - 2$, when initialized from configuration given by r-digit binary form of $R(L) = 2(L - 1)$. The other, based on 6-state cell, valid for $L > 2$, completes generation in $T = 3(L - 1)$ steps and requires initial configuration to be given by $R(L) = 2(L -2)$ in binary.

II. FORMAL DESCRIPTION OF THE CSG PROBLEM.

Let $CA = (C,\nu,\lambda)$ denote a cellular automaton, where $C = \{c_i\}$ is a set of cells, ν: $C \longrightarrow C^{\rho}$ is a neighborhood function and λ: $S^{\rho} \longrightarrow S$ is a local function of CA. The natural number $\rho > 1$ determines what is called a scope (index) of neighborhood.

By a cell $c = (S,S^{\rho-1},\lambda)$ the finite automaton is meant such that S is its finite, nonempty state set, $S^{\rho-1}$ denotes its input alphabet and λ is its transition function.

Only 1-D CAs are considered here, so the cells $c_i \in C$ are ordered according to their position $i \in Z$; Z is the set of integers. We assume in the paper that $\rho = 3$. Consequently for each $i \in Z$ we have $\nu(c_i) = (c_{i-1},c_i,c_{i+1})$ with left and right neighbors. The input alphabet $S^{\rho-1}$ of c_i is determined by the states of its neighbors.

Any 4-tuple (a,b,c,d) of elements from S for given CA, such that $\lambda(a,b,c) = d$ will be called the elementary rule (ER) of CA and will be denoted by abc/d; here the term "production" was also in use [4]. The local function λ of CA may be described then by the set of all its ERs. A state denoted by "$\cdot$" will be called the quiescent state with a property that $\cdot\cdot\cdot/\cdot$ is the ER of CA. A configuration of $CA = (C,\nu,\lambda)$ is defined as the function f. $C \longrightarrow S$, so S^C denotes the set of all configurations. Then, $f(\nu(c))$ is the state of neighborhood of given cell c. The configuration f is called finite if and only if $f(c_i) = \cdot$ for all but finitely many i's.

We shall represent finite configurations as follows. if $f \in S^C$ is such that for some i the equality $f(c_{i+j}) =$ holds for all $j < 0$ and for all $j > k - 1$ then f will be denoted by $\cdots w \cdots$ where $w = f(c_i)\ldots f(c_{i+k-1})$ and $k > 0$. The configuration f will be said to have the length k and to occur at the position i.

Suppose now that binary digits 0 and 1 are in state set S. Let $R(L) = aL + b$ be a linear polynomial with $R(L) > 0$ for all integers $L > 0$. Assume some $L > 0$ and $w = (R(L))_2$, where the binary form of $R(L)$ has r digits with 1 as the leftmost digit. Then the configuration

$\cdots w \cdots$ will be called either the simple representation of L (if $a = 1$ and $b = 0$) or the encoded representation of L, otherwise.

Let $x \in S$ and let $\cdots (x)^L \cdots$ denotes a configuration of L consecutive cells, all occuring in state x. It will be called the segment of length L.

By global transition function γ: $S^C \longrightarrow S^C$ of CA such a function is understood that $\gamma(f) = g$ implies $\gamma(f(c_i)) = \lambda(f(\nu(c_i))) = g(c_i)$ for all $i \in Z$. If $\gamma(f) = g$, then we say that CA passes from f to g.

Let $f_0 = \cdots w \cdots$ be some representation of given integer $L > 0$. If CA with global transition function γ passes from given initial f_0 to the final stable configuration $f_T = \cdots (x)^L \cdots$ in a such way that:

$\gamma(f_0) = f_1, \quad \gamma(f_1) = f_2, \quad \ldots, \quad \gamma(f_{T-1}) = f_T$

where $f_{i-1} \neq f_i$ for all $i = 1, 2, \ldots, T$, then we say that CA generates the segment of length L. T is the time of CSG process.

Now we can state the CSG problem: one might determine 'simple' $CA = (C,\nu,\lambda)$ capable to generate segments for any given length $L > 0$. Since $\rho = 3$ has been chosen, then S and λ rest to be searched for in the problem.

III. SOLUTION.

The solution presented here for the CSG behaviour assumes two levels of organization. On the higher level two groups of cells are distinguished in transient configurations, namely a counter and a sphere of two waves. Initial state of the counter is determined by $R(L)$. During generation process the states of counter are decremented until the zero state is reached. The role of counter is to control the waves. The emmision of waves occurs twice. at the beginning with the speed v, and at the end of counting with the speed 1/1, to assure that the waves can meet. We have.

Theorem 1. Let there be given CA – a solution of the CSG problem with a counter and two waves distinguished: a front one of speed v and a final one of speed 1/1. Let the representation of number L is determined by $R(L)$. Then: $T = L/v$ and $R(L) = L(1-v)/v$.

Proof. Is not presented here.

The lower level of organization (still above λ) with a set of 13 smaller components was also used in order to systematize computer aided searching of the solution. It is not explained here.

A number of 5-state solutions have been found. One of them is given in Table I. In *Fig.* 1. some parallelograms are shown; rows are shifted to have all counters synchronized. Following the general idea of solution the counter states are represented in transient configurations by RNS form. Thus the counter state number is created by either adding (symbol 1) or subtracting (symbol +) the proper weights according to i.

TABLE 1. Function λ for our solution ($h = 73$).

0	0	1	+	x	·
0	0	0	+	+	+
1	0	0	+		+
+			+	+	
x				·	
·			·	·	

1	0	1	+	x	·
0	1	1	0		0
1	1	1	0		0
+	1	1	0		
x			x	x	
·	1	1	·		

+	0	1	+	x	·
0	1	1	0	+	+
1	1	1	0	1	0
+			x	x	+
x				x	+
·			1	1	

x	0	1	+	x	·
0			+	+	
1			1	1	·
+			x	x	+
x	·	x	x	x	x
·		x		x	x

·	0	1	+	x	·
0					+
1					+
+					x
x					·
·	·	·	x	·	·

```
−5 ·····························101········
−4 ··················100······100+·······
−3 ············ ·11·····10++·····10++x······
−2 ······10·····10+····1+0+x····1+0x+······
−1 ·1····1++··  1++x····1+++·····1+++x·····
 0 ·0+····0+x···  0x+·····0x+x·····0xx+·····
 1 ··+x····++·   ··++x·····+x+·······+x+x····
 2 ·x1+···x1+x   x1x+····x1x+x····x1xx+····
 3 ·xx0x  ·xy1+···xx1+x  ··xx1x+····xx1x+x ··
 4 ·x·····xxx0x ·xxx1+···xxx1+x···xxx1x+···
 5 ·x·····xx···   xxxx0x··xxxx1+···xxxx1+x··
 6 ·x·····xx·····xxx·····xxxxx0x··xxxxx1+··
 7 ·x·····xx ····xxx·····xxxx·····xxxxxx0x·
 8 ·x·····xx····  xxx·····xxxx·····xxxxx····
 9 ·x·····xx·····xxx·····xxxx·····xxxxx····
L:   1,     2,      3,       4,        5
```

Fig.1. Parallelograms of the CSG processes.

Analysing the roles of all states in the frame of assumed idea for solution we may conclude its minimality.

Theorem 2. Assume a solution of the CSG problem as previously. Then a 5-state solution is the minimum state one.

Proof. Is not presented here.

In Table 2 we show some comlpexity factors of found CA; Q is a number of cells involved, h – a number of ERs and $i(\lambda) = 1-h/h\text{max}$ determines a logical circuit spare of λ.

TABLE 2. A comparison of some solutions.

sol.	$\lvert S \rvert$	ρ	T	L	$R(L)$	Q	h	$i(\lambda)$
[1]	8	3	$2(L-r)$	?	$L-r$	L	190	.63
[2]	7	3	$3L+r-2$	$L > 1$	$2L-2$	$L+r$	67	.80
[2]	6	3	$3(L-1)$	$L > 2$	$2L-4$	$L+r$	59	.73
	5	3	$2L+3$	$L > 0$	L	$L+r+2$	73	.42

REFERENCES.

[1] Blishun A.F. – Generating a cellular string of a given length (in Russ.), Izv. ANSSSR, Tekhn. Kib., 6, 1975, 95-98.

[2] Giorgadze A.H., Mandzgaladze P.W. Matevosjan A.A. – A way of generating a cellular string of a given length (in Russ.) Izv. ANSSSR, Tekhn. Kib., 1, 1985, 135-8.

[3] Hortensius P.D., McLeod R.D., Card H.C. – Parallel random number generation for VLSI systems using cellular automata. IEEE Tr. on Comp. C-38, 1989, 1466-73.

[4] Moore F.R., Langdon G.G. – A generalized firing squad problem. Inf. & Control, v.12, 1968, 212-20.

.5] Nourai F., Kashef R.S. – A universal four state cellular computer. IEEE Tr on Comp. C-24, 1975, 766-76.

[6] Serizawa T. - Three-state Neumann neighbor cellular automata capable of self-reproducing machines. Syst. Comp. Jpn. v.18, No.4, 1987, 33-40.

[7] Smith A.R. – Simple computation-universal cellular spaces. JACM, v.18, 1971, 339-53

[8] Steiglitz K., Kamal I, Watson A. – Embedding computations in one-dimensional automata by phase coding solitons. IEEE Tr. on Comp. C-37, 1988, 138-45.

COMPUTING HOMOMORPHISMS OF LABELLED DIRECTED GRAPHS IN PARALLEL USING HYPERCUBE MULTIPROCESSORS

BOLESLAW MIKOLAJCZAK
Computer and Information Science Department
Southeastern Massachusetts University
North Dartmouth, MA 02747, U. S. A.

Abstract -This paper deals with computing of vertex-edge homomorphisms of deterministic labelled directed graphs in parallel. A new algorithm is presented and its time computational complexity is evaluated. A decomposition of a graph with respect to different labels is used. Number of processors required during various stages of the algorithm is assessed.

I. INTRODUCTION

Traditionally, time and memory were the only resources considered when the computational complexity of an algorithm had to be evaluated. The architecture of the computing machine was a single processor sequentially addressing the memory cells. This was a reasonable model as long as processors are much more expensive than storage space. Recent developments in VLSI technology substantially reduced the processor-to-memory cost ratio. In this technology the cost of each feature is proportional to the area it consumes on the silicon wafer, and processors and memory cells have area of comparable size. The tradeoff between time and memory is extended to a time hardware tradeoff, where the hardware is a combination of processors and memory. Another justification for introducing parallelism has even deeper reason than technological innovations. We notice that the remedy which shortened the length of computation time, required by the sequential algorithm, was the introduction of parallelism. Clearly, every search problem is amenable to a single time-hardware tradeoff of the form $HT=N$. Simply partition the N points of the search space into H equal subsets, and assign a processor to search over the $T=N/H$ points of each subset. So, if parallelism is essential in overcoming some fundamental limitations of sequential algorithms, it is worthwhile to explore better ways of exploiting a multiprocessor system.

In parallel computations we have four major stages of algorithm development:
(i) choose the algorithm indicating the elementary computations and their interdependence
(ii) choose a particular multiprocessor architecture
(iii) find a schedule whereby the algorithm is executed on the processors (so that all necessary data are available at the appropriate processor at the time of each computation)
(iv) evaluate performance of the algorithm, measured as the makespan of the schedule.

In attacking various problems, two approaches seem natural the more practical approach is to insist on a polynomial bound on the number of processors, and then try to obtain the best time, perhaps the more theoretical approach is to insist on a polylog bound on the time, and then try to obtain the best processor count.

By an *efficient parallel algorithm* we mean one that takes polylogarithmic time using a polynomial number of processors. In practical terms, at most a polynomial number of processors is reckoned to be feasible. A polylogarithmic time algorithm takes $O(log^k n)$ parallel time for some constant integer k, where n is the problem size. Problems which can be solved within these constraints are universally regarded as having efficient parallel solutions and are said to belong to the class *NC*.

A subclass of problems of particular interest are those which have *optimal parallel algorithms.* An optimal parallel algorithm is an algorithm for which the product of the parallel time t with the number of processors p used is linear in the problem size n That is, $pt=O(n)$. Optimality may also mean that the product pt is equal to the computation time of the fastest known sequential-time algorithm for the problem. We specifically refer to the problem as having optimal speed-up.

For any problem for which there is no known polynomial-time sequential algorithm, for instance, any *NP complete problem* we cannot expect to find an efficient parallel solution using a polynomial number of processors. However, we might wish to find such a parallel solution for a problem with a polynomial-time sequential algorithm, i.e. a problem in class *P* . There are, however, many such problems which do not seem to admit parallelisation readily. These problems, which we refer to as being hardly parallelizable, form the class of *P complete problems* If an efficient parallel solution for any P complete problem could be found then a similar solution would exist for any other. There is no proof, but a great deal of circumstantial evidence, that classes *P* and *NC* differ.

All known sequential algorithms for NP-complete problems run in exponential time, and all known parallel algorithms have exponential cost For these problems three general solutions are in place. fast approximation algorithms, good probabilistic algorithms, and parallel approximation algorithms.

NP completeness of the automata homomorphism problem follows from considerations presented in Levin, Garey-Johnson, and Mikolajczak [3,6,8], and polynomial reducibility is from GRAPH K-COLORABILITY. All concepts not defined in this paper are taken from Mikolajczak [10] We assume that a concept of homomorphism discussed here includes transformations both on vertices (states) and edges (inputs) Such homomorphism is said to be the generalized homomorphism.

II. PARALLEL ALGORITHM

In this what follows we will apply the following assumptions.
(i) The available number of processors is adequate for dealing with the whole width of the directed acyclic graph (dag) which represents the algorithm (thus the number of processors involved is no longer a parameter).
(ii) A communication delay t between the time when some information is produced at a processor and the time it can be used by another processor is measured in elementary steps of the processors (or nodes of the dag), t is a parameter of the architecture.
(iii) The optimum makespan of the scheduling problem, being a function of t, is a fair measure of the parallel complexity of the dag, the scheduling problem is NP-complete.
(iv) We assume as a model of parallel computations the shared memory computer, in which a number of processors work together synchronously and communicate with a common random access memory, in the event of read or write conflicts in this shared memory we assume that both conflicts are allowed, and the lowest numbered processor succeeds in the case of a write conflict.

In our approach we apply a decomposition of domain and range of directed labelled graphs with respect to different labels. There exists also a second possibility to decompose directed labelled graphs into primaries (see Bavel [1]). Unfortunately, in the second case a decomposition of a graph is not a partition but a cover on a set of vertices, this makes load balancing more difficult (dynamic load balancing).

In algorithm description we apply the following notation.
$n_A=|S_A|$, $n_B=|S_B|$ number of vertices (states) of graph A and B, $m_A=|\Sigma_A|$, $m_B=|\Sigma_B|$ - number of different labels (inputs) of graph A and B, n-number of processors, $A_i=(S_A,\Sigma_{A,i},\delta_{A,i})$, $B_j=(S_B,\Sigma_{B,j},\delta_{B,j})$- autonomous labelled directed graphs A and B, respectively, where $1<=i<=m_A$, $1<=j<=m_B$, $\delta_{A,i}$ and $\delta_{B,j}$ are transition functions of graph A and B, respectively.

<u>SIMD parallel algorithm computing a set of generalized homomorphisms between deterministic complete labelled directed graphs</u>

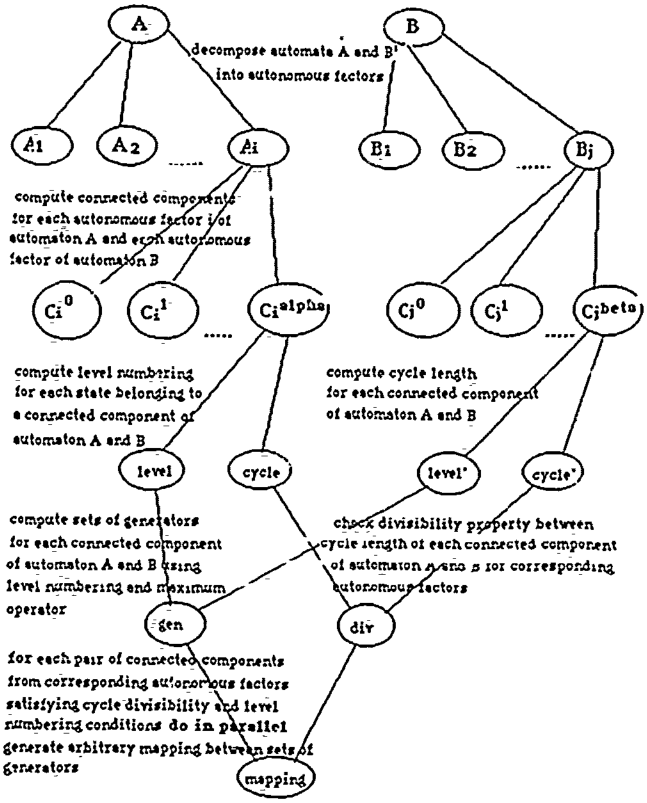
<u>Input:</u> Deterministic complete labelled directed graph $A=(S_A,\Sigma_A,\delta_A)$ and $B=(S_B,\Sigma_B,\delta_B)$ with disjoint vertex sets and edge sets.
<u>Output:</u> Set of all generalized homomorphisms from A to B denoted as GHom (A,B), i.e. set of state-input homomorphisms with transformations on set of vertices and on input semigroups.

Step 1: Decompose deterministic complete labelled graphs A and B into m_A and m_B autonomous factors, respectively; allocate these autonomous factors to m_A+m_B SIMD processors.
Step 2: For each autonomous factors of A_i and B_j, respectively, compute in parallel a set of all a_i connected components of A_i, and a set of all b_j connected components of B_j.
Step 3: For each connected component of A_i and B_j compute in parallel, using at most $a_i m_A+b_j m_B$ processors, the following topological characteristics:
a) lengths of cycles $d_i^{i'}$ for each connected component $C_i^{i'}$ of A_i, where $1<=i<=m_A$, and $1<=i'<=a_i$
b) lengths of cycles $d_j^{j'}$ for each connected component $D_j^{j'}$ of B_j, where $1<=j<=m_B$, and $1<=j'<=b_j$
c) level enumerations for each vertex belonging to every connected component $C_i^{i'}$ of autonomous factor A_i and $D_j^{j'}$ of autonomous factor B_j (substeps a,b,c are independent and can also be performed in parallel).
Step 4: For each connected component of A_i and B_j compute in parallel using at most $a_i m_A+b_j m_B$ processors:
(i) a set of generators using maximum function (a set of generators for connected component is defined as a set of all states of this component which have maximum value of vertex level enumeration, if a connected component is strongly connected then arbitrary state of this component can be treated as a generator)
(ii) check divisibility property of the lengths of cycles $d_j^{j'}$ of each connected component $D_j^{j'}$ of autonomous factor B_j with respect to the lengths of cycles $d_i^{i'}$ of each connected component $C_i^{i'}$ of autonomous factor A_i.
Step 5: For each pair of connected components $(C_i^{i'}, D_j^{j'})$ such that cycle length $d_j^{j'}$ divides $d_i^{i'}$ generate nondeterministically and in parallel using at most $a_i b_j m_A m_B$ processors a set of all possible mappings between set of generators of $C_i^{i'}$ and set of generators of $D_j^{j'}$.
Step 6: For each mapping computed in step 5 generate in parallel using at most $(m_B n_B)^{(m_A n_A)}$ processors a binary relation of successors implied by transition functions of connected components $C_i^{i'}$ and $D_j^{j'}$; this binary relation should be computed modulo length of cycle $d_j^{j'}$ of connected component $D_j^{j'}$.
Step 7: For all binary relations generated in Step 6 check in parallel using at most $(m_B n_B)^{(m_A n_A)}$ processors whether this relation is a function, if such relation is a function then this is a generalized homomorphism.

The dependency graph of this algorithm is presented on Fig.1. As an example we provide a pseudocode of the parallel connected components algorithm. Implementations of other steps of the algorithm are not provided because of the lack of space.

Parallel connected components algorithm:

```
Step 1.  Broadcast data to all processor nodes by distributing vertices
and edges evenly, i.e. $n/N$ vertices and $e/N$ edges among $N$
active processors.
Initialize each vertex to be the root of the tree which contains it.
Step 2. For each vertex i and j do in parallel
            If vertex i and vertex j are in processor
               then process edge (i,j)
                        Find the root of vertex i.
                        Find the root of vertex j.
                        Merge (union) tree for i and tree for j.
            If processor node is not a collector node
                then pass parent array and all unprocessed edges to
                appropriate nodes with minimal Hamming distance.
Step 3:  For collector nodes do in parallel
          Combine the information in the current parent array with the
          parent array passed to node from sender node.
Repeat steps 2 and 3 for all unprocessed edges and parent arrays.
```


A
B
decompose automata A and B
into autonomous factors
A1
A2
Ai
B1
B2
Bj
compute connected components
for each autonomous factor i of
automaton A and each autonomous
factor of automaton B
Ci0
Ci1
Cialpha
Cj0
Cj1
Cjbeta
compute level numbering
for each state belonging to
a connected component of
automaton A and B
compute cycle length
for each connected component
of automaton A and B
level
cycle
level'
cycle'
compute sets of generators
for each connected component
of automaton A and B using
level numbering and maximum
operator
check divisibility property between
cycle length of each connected component
of automaton A and B for corresponding
autonomous factors
gen
div
for each pair of connected components
from corresponding autonomous factors
satisfying cycle divisibility and level
numbering conditions do in parallel
generate arbitrary mapping between sets of
generators
mapping

III. REFERENCES

1. Bavel, Z., *Introduction to the Theory of Automata and Sequential Machines*, Science Research Associates, California, 1971
2. Chin F.Y., Lam J., Chen I-N., *Efficient parallel algorithms for some graph problems*, Communications of the ACM, vol.25, 9(1982), pp. 659-665.
3. Garey, M. R., Johnson D. S., *Computers and Intractability. A Guide to the Theory of NP-completeness*, W.H. Freeman and Company, New York, 1979, pp.202-203.
4. Grzymala-Busse, J.W., *Operation preserving functions and autonomous factors of finite automata*, Journal of Computer and System Sciences, 5(1971), pp.465-474.
5. Hirschberg D.S., Chandra A.K., Sarwate D.V., *Computing connected components on parallel computers*, Communications of the ACM, vol.22,8(1979), pp.461-464.
6. Levin, L. A., *Universal sorting problems*, Problemy Peredaci Informacii, 9(1973), pp.115 116, (in Russian), English translation in "Problems of Information Transmission", 9, pp. 265-266.
7. Mikolajczak, B., *Generalized functions preserving operations of finite automata*, Foundations of Control Engineering, 6(1981), pp.211-247.
8. Mikolajczak, B., *Time computational complexity of some decision and search problems in finite automata*, in Methods of Operations Research, Athenaum Scriptor, Augsburg, vol.43, (1981), pp.405 417.
9. Mikolajczak, B., *Transformations and Computational Complexity of Problems in Automata Theory*, Polish Academy of Sciences. Warsaw-Poznan, pp.1-90, 1988 (in Polish).
10. Mikolajczak, B., (ed.), *Algebraic and Structural Automata Theory*, in series Advances in Discrete Mathematics, Vol.44, North Holland Publishing Company, pp.1-402, (1991).
11. Moitra A., Iyengar S. S., *Parallel algorithms for some computational problems*, in. Advances in Computers, vol.26, (1987), pp.93-153.
12. Papadimitriou, Ch., H., Yannakakis M., *Towards an architecture-independent analysis of parallel algorithms* (extended abstract), Proc. of the ACM Symposium on Foundations of Computer Science, 1988, pp. 510-513.

LU DECOMPOSITION ON A SHARED MEMORY MULTIPROCESSOR

Paul A. Farrell
Dept. of Mathematics & Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Arden Ruttan
Dept. of Mathematics & Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Abstract:- We propose an algorithm for the parallel LU decomposition of an upper Hessenberg matrix on a shared memory multiprocessor. We consider the general case of p processors, where p is not related to the size of the matrix problem. We show that the LU decomposition of an $(m+1)$-banded Hessenberg matrix can be achieved in $O(\frac{3nm^2}{p})$ operations, where n is the dimension of the matrix and p is the number of processors. For tridiagonal matrices this algorithm has a lower operation count than those in the literature and yields the best existing algorithm for the solution of tridiagonal systems of equations.

1. Introduction

A number of authors over the last two decades have written on parallel algorithms for solving tridiagonal systems. These articles have considered the problem of solving tridiagonal systems for the form $Ax = b_i$, $1 \le i \le k$ where A and all of the b_i, are known at the start of the process. In such cases, the computations can be arranged to produce highly efficient parallel solutions to all m systems simultaneously. It should be noted, however, that there are a number of common numerical situations, for example the ADI method, where one needs to solve tridiagonal systems where A is known *ab initio* but the b_i's are not all known at the start of the computation but rather arise as a result of an iteration process.

2. LU Decomposition Algorithm

We shall, in fact, consider the LU decomposition of an $n \times n$ upper Hessenberg matrix, since the analysis is not significantly more difficult and the additional generality leads to insights, which produce a more efficient algorithm. Let $A = (a_{ij})$ be a banded $n \times n$ upper Hessenberg matrix with band width $m+1$, i.e., $a_{ij} \neq 0$ only when $\min\{1, i-1\} \le j \le \max\{n, m+i-1\}, 1 \le i \le n$. It suffices to consider the case where $a_{i+1,i} \neq 0, 1 \le i \le n-1$, since otherwise the matrix is reducible, and we may consider the LU decomposition of the subproblems resulting from the reduction. Throughout this paper we will use the convention that any element with a nonpositive index has value zero.

As in most algorithms for shared memory multiprocessors, the object here is to partition the problem into a number of subproblems suitable for solution by tasks running on the available processors. We shall consider the general case of p processors, where p is not related to the size of the matrix problem. Clearly A has an LU factorization, $A = LU$, where L is a unit lower bi-diagonal $n \times n$ matrix and $U = (u_{ij})$ is a banded $n \times n$ upper triangular, with m non zero diagonals, including the main diagonal. The special form of L allows one to readily determine L^{-1}. One finds that $L^{-1} = (\hat{\ell}_{ij})$ is an $n \times n$ lower triangular matrix given by

$$\hat{\ell}_{ij} := \begin{cases} \prod_{t=j+1}^{i} (-\ell_t) & i \ge j \\ 0 & i < j. \end{cases} \qquad (1)$$

Thus the elements of $U = L^{-1}A$, satisfy $1 \le i,j \le n$

$$u_{ij} = \sum_{s=j-m+1}^{\min\{i,j+1\}} \hat{\ell}_{is} a_{sj} = \sum_{s=j-m+1}^{\min\{i,j+1\}} \prod_{t=s+1}^{i} (-\ell_t) a_{sj}. \qquad (2)$$

As in the tridiagonal case, the well known substitution (*cf.* [2], pp. 473 - 474)

$$\begin{aligned} &y_1 = 1 \\ &\ell_i = y_{i-1}/y_i \qquad i = 2,3,\cdots,n \end{aligned} \qquad (3)$$

can be used to simplify (2). Since $u_{j+1,j} = 0$, for $1 \le j \le n-1$, (2) yields the following linear systems for the unknowns y_i.

$$\begin{aligned} &y_j = 0, \;\; j \le 0 \quad , \quad y_1 = 1 \\ &y_{j+1} a_{j+1,j} = \sum_{s=j-m+1}^{j} (-1)^{j-s} y_s a_{sj}, \quad 1 \le j \le n-1. \end{aligned} \qquad (4)$$

It is clear that (4) defines an $m+1$ banded triangular linear system

$$Ty = e. \qquad (5)$$

where

$$t_{ij} = \begin{cases} 1 & \text{if} \;\; i = j = 1 \\ (-1)^{i-j} a_{j,i-1} & j \le i \;,\; i > 1 \\ 0 & j > i \end{cases}$$

Thus the problem of finding an LU factorization of an upper Hessenberg matrix reduces to solving the banded triangular system described in (5) to obtain the y_i's and then using the solution of that system to evaluate L and U. In practice, L may be determined from equation (3). To determine U, note first that the elements of the $(m-1)^{st}$ diagonal, $u_{i,i+m-1}$ satisfy $u_{i,i+m-1} = a_{i,i+m-1}$, for $i = 1,\ldots,n-m+1$. Also $u_{1,j} = a_{1,j}$ for $j = 1,\ldots,m$. Thus these elements do not require any calculation. The j^{th} super-diagonal is given in terms of the $(j+1)^{st}$ by

$$\ell_i u_{i-1,j+i} + u_{i,j+i} = a_{i,j+i}, i = 2,\ldots,n-j. \qquad (6)$$

Note that each element depends only on a single element of the next superdiagonal and on known values from L and A. Thus, the calculation of each super diagonal of U is perfectly parallelizable. In fact, the main diagonal may be obtained using 1 division rather than the multiplication and subtraction in (6) by

$$u_{i,i} = a_{i+1,i}/\ell_{i+1}, i = 2,\ldots,n. \qquad (7)$$

Further, the calculation of the super-diagonals can be chained.

Thus the calculation of L using (3) requires $n-1$ divisions. The total computations for U is

$$(n-1) + 2\sum_{j=1}^{m-2} (n-j-1) = n(2m-3) - (m^2 - m - 1). \qquad (8)$$

The latter term is negative for $m \ge 1$. Hence we get the following upper bound for the complexity of calculating U

$$n(2m-3).$$

Note that in the case of a tridiagonal system, $m = 2$, (8) reduces exactly to $n-1$. Hence using p processors L and U can be calculated in the general case in

$$\frac{2n(m-1)}{p} \qquad (9)$$

operations and in the special tridiagonal case in

$$2(n-1)/p \qquad (10)$$

operations. There remains only the solution of the triangular system $Ty = e$.

3. Algorithm for the Triangular System

In [3], Lakshmivarahan and Dhall present an algorithm for calculating the LU factorization of a tridiagonal matrix. Their algorithm used the substitution given in (3) to produce a linear system, which is equivalent to that described by (5) with $m = 2$. Our algorithm is a generalization to the upper Hessenberg case of the algorithm presented in [3]. We remark that the improvement produced in the generalization leads also to a more efficient algorithm for the tridiagonal case.

In order to partition the problem we set

$$z_j = (y_{j-m+1}, y_{j-m+2}, \cdots, y_j)^T, \quad 1 \leq j \leq n,$$

and let $B_j, 1 \leq j \leq n-1$, be the $m \times m$ matrix

$$B_j := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & 0 & 1 \\ b_1^j & \cdots & & b_{m-1}^j & b_m^j \end{bmatrix} \tag{11}$$

where $b_i^j := (-1)^{m-i} a_{j-m+i,j}/a_{j+1,j} \quad j = 1,2,\cdots,n-1$. We obtain from (4) the m-vector iteration

$$z_{j+1} = B_j z_j \quad j = 1,2,\cdots,n-1. \tag{12}$$

In order to evaluate $y_1, y_2, \quad , y_n$, one must compute

$$z_{km+1} = \left(\prod_{i=1}^{k} C_i\right) z_1, \quad k = 1,2,3,\cdots,N := \lceil (n-1)/m \rceil. \tag{13}$$

where $C_i := \prod_{j=(i-1)m+1}^{im} B_j$, $1 \leq i \leq N-1$ and $C_N := \prod_{j=(N-?)m+1}^{n-1} B_j$.

To calculate the required products $\prod_{i=1}^{k} C_i$, $k = 1,2,\cdots,N$, one uses a variant of recursive doubling. Let $Z_1 = C_1 z_1$ and $Z_i = C_i$, $i = 2,\ldots,N$, then, assuming we have g processor groups, each group first calculates

$$D_{l,k} = \prod_{i=(k-1)M+2}^{(k-1)M+l} Z_i, \; k = 1,\ldots,g, \; l = 2,\ldots,M = N/g.$$

Then the g processor groups execute the following algorithm:

```
for $i := 0$ thru $log(g) - 1$ do
{distribute the g/2 independent calculations found in the}
{ $j$ and $k$ loops below among the g/2 groups of processors }
for $j := 2^i$ thru $g - 2^i$ step $2^{i+1}$ do
    for $k := j + 1$ thru $j + 2^i$ do
        {using 2 groups calculate }
        for $l = 1$ thru $M$ do
            $D_{l,k} := D_{l,k} D_{M,j}$
```

It is easily seen that after the execution of the above algorithm $D_{l,k} = (\prod_{j=1}^{(k-1)M+l} C_j) z_1$. For the purpose of simplifying the complexity analysis, assume that $n = Nm + 1$, $p = g(2m-1)$ where g is a power of 2, and $N = gM$.

An analysis, the details of which appear in [1], yields the following time complexity estimate. For a general upper Hessenberg matrix with band width $m + 1$, the time required to calculate its y_i's in this fashion is

$$\frac{n}{2p}\left[6m^2 - 2m + 1 + m(2m-1)\log\left(\frac{p}{2m-1}\right)\right] + O(m). \tag{14}$$

In the tridiagonal case $m = 2$, and this reduces to

$$\frac{n}{p}\left[\frac{21}{2} + 3\log(\frac{p}{3})\right]. \tag{15}$$

Thus, from (9) and (14), the total cost of an LU factorization is

$$\frac{n}{2p}\left[6m^2 + 2m - 3 + (2m^2 - m)\log\left(\frac{p}{2m-1}\right)\right] + O(m). \tag{16}$$

In the tridiagonal case, by (10) and (15), the cost of producing an LU factorization is

$$\frac{n}{p}\left[\frac{25}{2} + 3\log\left(\frac{p}{3}\right)\right] + O(1). \tag{17}$$

If one's goal is to solve the linear system $Ax = b$, in addition to solving (5), one must also perform the backsolve by solving the banded triangular systems $Lz = b$ and $Ux = z$. In the tridiagonal case, this may be done by casting it as the solution of two linear recurrences, similar to (2.3) and (2.4) in [3]. The recurrences may then be cast in the form $z_i = \alpha_i z_{i-1} + \beta_i$ and solved using *Algorithm A* and *Algorithm Y* from [3]. The complexity involved, in the tridiagonal case, is $2n/p$, to cast the analogue of (2.3) in [3] in the appropriate form, and $3n(2 + \log(2p/3))/p$ to solve the two recurrences, using *Algorithm A* and *Algorithm Y*. Adding these gives a total of

$$\frac{n}{p}(8 + 3\log(\frac{2p}{3})). \tag{18}$$

The cost of solving for one righthand side, given by (17) and (18), is thus

$$\frac{n}{p}\left[\frac{47}{2} + 6\log\left(\frac{p}{3}\right)\right] + O(1). \tag{19}$$

4. Conclusions

In the tridiagonal case, the algorithm is not only better than existing algorithms in the literature for LU decomposition, but also has better computational complexity for the solution of a single tridiagonal system, as indicated in Table 1, where $n' = n + 1 = 2^t$.

Method	Processors	Time
Serial Gaussian Elimination	1	8n
Recursive Doubling [2]	n	$24\log n$
Odd-Even Reduction [2]	$n'/2$	$19\log n' - 14$
Odd-Even Elimination [2]	n'	$14\log n' + 1$
Lakshmivarahan Dhall [3]	$n/2$	$18\log n$
Lakshmivarahan Dhall [3]	p $3 \leq p \leq \frac{3n}{4}$	$(n/p)[25 + 9\log p/3] - 3$
Algorithm	p	$(n/p)[47/2 + 6\log p/3]$

Table 1: Complexity of the solution of a single linear system for tridiagonal matrices.

Further let us consider again cases, such as the ADI method discussed in the introduction, where A is known in advance but the b_i are not. Comparing this algorithm with methods, such as Recursive-Doubling, which do not perform the LU decomposition, a further improvement in computational efficiency results, since one need only perform the forward and back solves, for each right-hand side, rather than performing the full elimination.

References

[1] J.Buoni, P.A. Farrell, A. Ruttan, *Algorithms for LU Decomposition on a Shared Memory Multiprocessor*, Technical Report CS-90-10-28, Department of Mathematics and Computer Science, Kent State University, 1990.

[2] R.W. Hockney, C.R. Jesshope, *Parallel Computers 2 - Architecture, Programming and Algorithms* (Hilger, Bristol, 1988).

[3] S. Lakshmivarahan, S.K. Dhall, A New Class of Parallel Algorithms for Solving Tridiagonal Systems, *IEEE Fall Joint Computer Conference* (1986) 315-324.

[4] A.H. Sameh, R.P. Brent, Solving Triangular Systems on a Parallel Computer, *SIAM JNA* 14(6) (1977) 1101-1113.

A BICONJUGATE GRADIENT TYPE ALGORITHM ON MASSIVELY PARALLEL ARCHITECTURES *

Roland W. Freund
RIACS, Mail Stop Ellis Street
NASA Ames Research Center
Moffett Field, CA 94035, U.S.A.

AND

Marlis Hochbrück
Institut für Praktische Mathematik
Universität Karlsruhe
Englerstraße 2
D-7500 Karlsruhe, F.R.G.

Abstract — The biconjugate gradient (BCG) method is the "natural" generalization of the classical conjugate gradient algorithm for Hermitian positive definite matrices to general non-Hermitian linear systems. Unfortunately, the original BCG algorithm is susceptible to possible breakdowns and numerical instabilities. Recently, Freund and Nachtigal have proposed a novel BCG-type approach, the quasi-minimal residual method (QMR), which overcomes the problems of BCG. Here, we present an implementation of QMR based on an s-step version of the nonsymmetric look-ahead Lanczos algorithm. The main feature of the s-step Lanczos algorithm is that, in general, all inner products, except for one, can be computed in parallel at the end of each block; this is unlike the standard Lanczos process where inner products are generated sequentially. The resulting implementation of QMR is particularly attractive on massively parallel SIMD architectures, such as the Connection Machine.

INTRODUCTION

We are concerned with the iterative solution of large sparse linear systems

$$Ax = b, \tag{1}$$

where A is a nonsingular, in general non-Hermitian $N \times N$ matrix. Some ot the most efficient iterative schemes for (1) are *Krylov subspace methods:* for any initial guess $x_0 \in \mathbf{C}^N$, they generate approximations to $A^{-1}b$ of the form

$$x_n \in x_0 + K_n(r_0, A), \quad n = 1, 2, \ldots, \tag{2}$$

where $r_0 = b - Ax_0$ and

$$K_n(r_0, A) = \operatorname{span}\{r_0, Ar_0, \ldots, A^{n-1}r_0\} \tag{3}$$

is the nth *Krylov subspace* generated by r_0 and A. For example, the generalized minimal residual algorithm (GMRES) of Saad and Schultz [8] and the biconjugate gradient algorithm (BCG) of Lanczos [6] both satisfy (2). Unfortunately, for methods like GMRES, work and storage requirements per iteration grow linearly with n and, therefore, versions with restarts are used in practice, which often results in slow convergence. In contrast, for BCG, work and storage requirements per iteration are constant and low. However, BCG typically exhibits a rather irregular convergence behavior and the method can even break down.

THE QMR APPROACH

In [3], Freund and Nachtigal have proposed a BCG-type approach, the quasi-minimal residual algorithm (QMR), which overcomes the problems of BCG. The method uses an implementation developed by Freund, Gutknecht, and Nachtigal [1, 2] of the nonsymmetric Lanczos algorithm [5] with look-ahead [7] to generate basis vectors $v_1, v_2, \ldots$ for the Krylov subspaces (3). More precisely, with

$$\begin{aligned} V^{(n)} &= [v_1\ v_2\ \cdots\ v_n] = [V_1\ V_2\ \cdots\ V_l], \\ & V_k = [v_{n_k}\ v_{n_k+1}\ \cdots\ v_{n_{k+1}-1}], \quad k = 1, \ldots, l = l(n), \end{aligned} \tag{4}$$

we have

$$K_n(r_0, A) = \left\{ V^{(n)}z \mid z \in \mathbf{C}^n \right\} \quad \text{for} \quad n = 1, 2, \ldots . \tag{5}$$

The blocks V_k in (4) just contain the vectors corresponding to the kth look-ahead Lanczos step of length

$$h_k = n_{k+1} - n_k.$$

In the sequel, we refer to the first vectors v_{n_k} in each block as *regular vectors*, while the remaining vectors are called *inner vectors*. Furthermore, the relation

$$AV^{(n)} = V^{(n+1)}H^{(n)} \tag{6}$$

holds. Here $H^{(n)}$ is an $(n+1) \times n$ upper Hessenberg matrix which is also block tridiagonal with l diagonal blocks of size $h_k \times h_k$, $k = 1, 2, \ldots, l$. In addition to the *right Lanczos vectors* $v_1, v_2, \ldots$, the look-ahead Lanczos algorithm generates *left Lanczos vectors* $w_1, w_2, \ldots$ such that

$$K_n(w_1, A^T) = \operatorname{span}\{w_1, w_2, \ldots, w_n\} \quad \text{for} \quad n = 1, 2, \ldots ,$$

and, as in (4), we set

$$W_k = [w_{n_k}\ w_{n_k+1}\ \cdots\ w_{n_{k+1}-1}], \quad k = 1, \ldots, l.$$

These vectors are just constructed such that right and left Lanczos vectors corresponding to different look-ahead steps are biorthogonal, *i.e.*,

$$W_j^T V_k = \begin{cases} 0 & \text{if } j \neq k, \\ D_k & \text{if } j = k, \end{cases} \quad j, k = 1, \ldots, l, \tag{7}$$

and, moreover, the matrices D_k are all nonsingular.

By means of (5) and (6), the nth iterate (2) of any Krylov subspace method and the corresponding residual vector can be written as follows:

$$\begin{aligned} x_n &= x_0 + V^{(n)}z_n \quad \text{for some} \quad z_n \in \mathbf{C}^n, & (8) \\ r_n &= b - Ax_n = V^{(n+1)}\left(\|r_0\|_2 e_1 - H^{(n)}z_n\right). & (9) \end{aligned}$$

Here e_1 denotes the first unit vector in $\mathbf{R}^{n+1}$.

For the QMR method the parameter vector z_n in (8) is chosen such that the Euclidean norm of the coefficient vector in the representation (9) is minimal, *i.e.*, as solution of the least squares problem

$$\min_{z \in \mathbf{C}^n} \left\| \Omega_n \left(\|r_0\|_2 e_1 - H^{(n)}z \right) \right\|_2, \tag{10}$$

where $\Omega_n = \operatorname{diag}(\|v_1\|_2, \|v_2\|_2, \ldots, \|v_{n+1}\|_2)$. Here, Ω_n is chosen such that all basis vectors $v_j/\|v_j\|_2$, $j = 1, \ldots, n+1$, in the representation (9) of r_n have the same Euclidean length. Note that $\Omega_n H^{(n)}$ is an upper Hessenberg matrix with full column rank n. Hence (10) always has a unique solution z_n and the QMR iterate x_n is well defined by (8) and (10). Finally, we remark that z_n can be easily updated from step to step, and the resulting QMR algorithm can be implemented using only short recurrences (see [3] for details).

AN s-STEP LANCZOS ALGORITHM WITH LOOK-AHEAD

To enforce the biorthogonality conditions (7), inner products of vectors of length N need to be computed. In the implementation of the look-ahead Lanczos algorithm described in [1, 2], this is done sequentially, *i.e.* inner products are calculated in each iteration step n. On a massively parallel machine, such as the Connection Machine, the sequential computation of these inner products represents a bottleneck.

In this section, we sketch a version of the look-ahead Lanczos algorithm which overcomes this problem and is more suited for a parallel machine. In contrast to the sequential algorithm, where look-ahead

* This work was supported in part by Cooperative Agreement NCC 2 387 between NASA and the Universities Space Research Association (USRA)

steps of size $h_k > 1$ are performed only if necessary to avoid break downs of the Lanczos process, the philosophy of the s-step Lanczos algorithm is to construct Lanczos blocks of given size $h_k = s$, when ever possible. This is done by first generating $s - 1$ intermediate inner vectors by means of simple three-term recurrences

$$\begin{aligned} \tilde{v}_{n+1} &= A\tilde{v}_n - \zeta_n\tilde{v}_n - \eta_n\tilde{v}_{n-1}, & (11)\\ \tilde{w}_{n+1} &= A^T\tilde{w}_n - \zeta_n\tilde{w}_n - \eta_n\tilde{w}_{n-1}; & (12) \end{aligned}$$

with suitably chosen coefficients ζ_n, η_n, and $\eta_{n_k} = 0$. The biorthogonality conditions (7) are then enforced only at the end of each block. This has the advantage that all inner products arising in the biorthogonalization process for the inner vectors of a whole block can be computed in parallel. We remark that to enforce (7) for the inner vectors in block l, it is sufficient to biorthogonalize them only against the vectors from the previous blocks $f = f(n), f+1, \ldots, l$ using

$$\begin{aligned} v_n &= \tilde{v}_n - V_f D_f^{-1} W_f^T \tilde{v}_n - \ldots - V_{l-1} D_{l-1}^{-1} W_{l-1}^T \tilde{v}_n & (13)\\ w_n &= \tilde{w}_n - W_f D_f^{-T} V_f^T \tilde{w}_n - \ldots - W_{l-1} D_{l-1}^{-T} V_{l-1}^T \tilde{w}_n. & (14) \end{aligned}$$

Moreover, in general, only one previous block occurs in (13) and (14), *i.e.*, $f = l - 1$.

In [4], Kim and Chronopoulos proposed an s-step Lanczos algorithm using a fixed block size s throughout the whole process. Our numerical tests show that such an approach is not viable. In order to obtain a robust implementation of the s-step Lanczos algorithm, it is crucial to keep the block size variable and combine the process with a suitable look-ahead strategy.

In the following algorithm, we outline the s step look ahead Lanczos method which we propose. In each block step, the algorithm tries to build a block of size s. If the construction of such a block would lead to a singular or a nearly singular matrix D_l in (7) or to a new pair $v_{n_{l+1}}$ and $w_{n_{l+1}}$ of regular vectors which have dominant components in the old Krylov subspaces $K_{n_l}(v_1, A)$ or $K_{n_l}(w_1, A^T)$, we either build a smaller block or, by performing sequential steps, a bigger block

Algorithm. *Sketch of $s-$ step Lanczos algorithm with look-ahead*

```
0) Set $v_1 = r_0/\|r_0\|_2$ and choose $w_1 \in \mathbb{C}^N$ with $\|w_1\|_2 = 1$,
   Set $n_1 = 1$, $l = 1$, $\tilde{v}_0 = \tilde{w}_0 = 0$;
For $l = 1, 2, \ldots$:

   1) Compute $s - 1$ intermediate inner vectors via (11) and (12) for
      $n = n_l, \ldots, n_l + s - 2$;
      Set $\tilde{V}_l = [\tilde{v}_{n_l} \cdots \tilde{v}_{n_l+s-1}]$, $\tilde{W}_l = [\tilde{w}_{n_l} \cdots \tilde{w}_{n_l+s-1}]$;

   2) Construct the symmetric matrix $\tilde{W}_l^T \tilde{V}_l$;

   3) (Biorthogonalization of inner vectors.)
      Determine $f$ by $n_f = \max\{n_j \mid n_j \le n_l - s + 1\}$.
      For $n = n_l + 1, \ldots, n_l + s - 1$, compute $v_n$ and $w_n$ via (13) and
      (14); If $\|v_n\|_2 = 0$ or $\|w_n\|_2 = 0$, stop;
      Set $V_l = [v_{n_l} \cdots v_{n_l+s-1}]$, $W_l = [w_{n_l} \cdots w_{n_l+s-1}]$;

   4) Construct the symmetric matrix $D_l = W_l^T V_l$;

   5) Decide whether to construct $v_{n_l+s}$ and $w_{n_l+s}$ as regular vectors
      or to reduce the block size and go to 8) or 6), respectively;

   6) If it is possible to construct regular vectors $v_{n_l+\underline{s}}$ and $w_{n_l+\underline{s}}$ for
      $\underline{s} < s$:
      set $n_{l+1} = n_l + \underline{s}$, $V_l = [v_{n_l} \cdots v_{n_l+\underline{s}-1}]$, $W_l = [w_{n_l} \cdots w_{n_l+\underline{s}-1}]$,
      and go to 8);
      Otherwise, try to increase the block size $s$ by sequential steps:
      set $\bar{s} = s$;
      Loop:
      Set $\bar{s} = \bar{s} + 1$, $n = n_l + \bar{s} - 2$, compute $\tilde{v}_{n+1}$ and $\tilde{w}_{n+1}$ via (11)
      and (12), and biorthogonalize immediately:
      determine $f$ by $n_f = \max\{n_j \mid n_j \le n_l - \bar{s} + 1\}$ and compute
      $v_{n+1}$ and $w_{n+1}$ using formulas (13) and (14) (with $n$ replaced by
      $n + 1$); If $\|v_{n+1}\|_2 = 0$ or $\|w_{n+1}\|_2 = 0$, stop;
      Set $V_l = [V_l\ v_{n+1}]$, $W_l = [W_l\ w_{n+1}]$ and update the matrix $W_l^T V_l$.
      This loop is terminated if we can construct regular vectors $v_{n_l+\bar{s}}$
      and $w_{n_l+\bar{s}}$ or if we have reached the maximum block size. In the
      first case, go to 8), in the second case, go to 7);

   7) Determine the smallest value which failed the checks and up-
      date the upper bound $n(A)$ to this value. The block is now
      enforced to close. Let its size be $\bar{s}$ and set $n_{l+1} = n_l + \bar{s}$,
      $V_l = [v_{n_l} \cdots v_{n_l+\bar{s}-1}]$ $W_l = [w_{n_l} \cdots w_{n_l+\bar{s}-1}]$;

   8) (Construct regular vectors $v_{n_{l+1}}$ and $w_{n_{l+1}}$.)
      Set $n = n_{l+1}$, $\tilde{v}_n = A\tilde{v}_{n-1}$, $\tilde{w}_n = A^T\tilde{w}_{n-1}$, and compute

         $\tilde{v}_n = \tilde{v}_n - V_{l-1} D_{l-1}^{-1} W_{l-1}^T \tilde{v}_n - V_l D_l^{-1} W_l^T \tilde{v}_n$,
         $\tilde{w}_n = \tilde{w}_n - W_{l-1} D_{l-1}^{-T} V_{l-1}^T \tilde{w}_n - W_l D_l^{-T} V_l^T \tilde{w}_n$;

      If $\|\tilde{v}_n\|_2 = 0$ or $\|\tilde{w}_n\|_2 = 0$, stop;
      Otherwise, set $v_n = \tilde{v}_n/\|\tilde{v}_n\|_2$ and $w_n = \tilde{w}_n/\|\tilde{w}_n\|_2$;

   9) Construct the $l$th blocks of the block tridiagonal matrix $H^{(n-1)}$
      and set $l = l + 1$.
```

We note that the quantity $n(A)$ in step 7) is an estimate of the norm of the matrix A which is used for our checks to guarantee that the Lanczos vectors remain sufficiently linearly independent. A similar concept was first introduced for the sequential look-ahead Lanczos algorithm in [1]. These checks, the criteria for the decision in step 5), and further details of the algorithm will be presented in a forthcoming paper. Here, we only remark that the important properties (5), (6), and (7), which were used in the derivation of the QMR method, remain valid also for the s-step Lanczos algorithm with look ahead. Also, we note that the above algorithm can be realized with the same number of inner products as in the classical nonsymmetric Lanczos method without look-ahead. In particular, the $s \times s$ matrix $\tilde{W}_l^T \tilde{V}_l$ in step 2) can be constructed by computing only $2s - 1$ inner products, rather than s^2 as the straightforward approach would suggest. Moreover, in step 4), the matrix D_l can be updated from $\tilde{W}_l^T \tilde{V}_l$, using only already available inner products. Finally, numerical experiments with an implementation of the resulting QMR algorithm on the CM 2 will be reported elsewhere.

References

[1] R.W. Freund, M.H. Gutknecht, and N.M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, Part I*, Technical Report 90.45, RIACS, NASA Ames Research Center, November 1990.

[2] R.W. Freund and N.M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, Part II*, Technical Report 90.46, RIACS, NASA Ames Research Center, November 1990.

[3] R.W. Freund and N.M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Technical Report 90.51, RIACS, NASA Ames Research Center, December 1990.

[4] S.K. Kim and A.T. Chronopoulos, *An efficient nonsymmetric Lanczos method on parallel vector computers*, Technical Report 90-38, University of Minnesota, July 1990.

[5] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand. 45, 255-282 (1950).

[6] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Natl. Bur. Stand. 49, 33-53 (1952).

[7] B.N. Parlett, D.R. Taylor, and Z.A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp. 44, 105 124 (1985).

[8] Y. Saad and M.H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. 7, 856-869 (1986).

FAST PARALLEL SOLUTION OF SPARSE TRIANGULAR SYSTEMS

Fernando L. Alvarado*
The University of Wisconsin—Madison
Madison, Wisconsin 53706, USA

Robert Schreiber†
RIACS
MS T045-1, NASA Ames Research Center
Moffett Field, California 94035

Abstract We consider parallel solution of a sparse system $Lx = b$ with triangular matrix L, which is often a performance bottleneck in parallel computation. When many systems with the same matrix are to be solved, we can improve parallel efficiency by representing the inverse of L as a product of a few sparse factors. We construct the factorization with the smallest number of factors, subject to the requirement that no new nonzero elements are created. Applications are to iterative solvers with triangular preconditioners, to structural analysis, or to power systems applications. Experimental results on the Connection Machine show the method to be highly valuable.

I. INTRODUCTION

There are two possible approaches to the parallel solution of triangular systems of equations. The usual approach is to exploit whatever parallelism is available in the usual substitution algorithm [4]. The second, which requires preprocessing, works with some representation of L^{-1}.

If L is sparse, its inverse is usually much denser. Here we consider a factorization $L^{-1} = \prod_{k=1}^{m} Q_k$ with sparse factors. Such a factorization is possible in which the factors have no more nonzeros than L [2]. The chief advantage of a factorization of L^{-1} is that all the necessary multiplications for the computation of $Q_k x$ can be performed concurrently. Thus, it is possible to take advantage of more parallelism in the solution of these equations.

We review the use of partitioned inverses of L. Any triangular matrix L can be expressed as a product of elementary matrices: $L = L_1 L_2 \cdots L_{n-1}$. The factor L_j is unit lower triangular and nonzero below the diagonal only in column j, *i.e.* it is elementary lower triangular.

Regrouping, we may write

$$L = \prod_{k=1}^{m} P_k \qquad (1)$$

where $P_k = L_{e_{k-1}+1} L_{e_{k-1}+2} \cdots L_{e_k}$ and

$$0 = e_0 < e_1 < \cdots < e_m = n - 1 . \qquad (2)$$

Here $\{e_k\}_{k=0}^{m}$ is a monotonically increasing integer sequence. The factor P_k is lower triangular and is zero below its diagonal in all columns except columns $e_{k-1} + 1$ through e_k, where it is identical to L.

The solution of the partitioned problem proceeds as follows. From (1) it follows that

$$x = L^{-1} b = \prod_{k=m}^{1} P_k^{-1} b . \qquad (3)$$

In computing the matrix-vector products, we may exploit parallelism fully, using as many processors as there are nonzeros in P_k and summing the results in logarithmic time.

A. Problems Addressed

We say that the matrix X is *invertible in place* if $x_{ij} \neq 0 \Leftrightarrow (X^{-1})_{ij} \neq 0$. The elementary lower triangular matrices are invertible in place. There therefore always is at least one partition (1) of L with factors that invert in place.

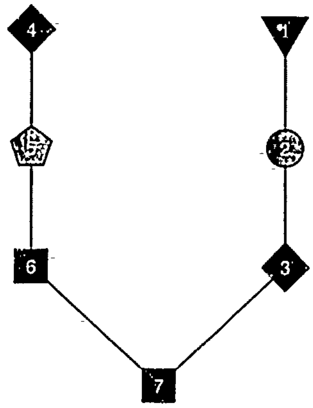


Figure 1: Best no-fill partition for a graph without reordering. Five factors are required.

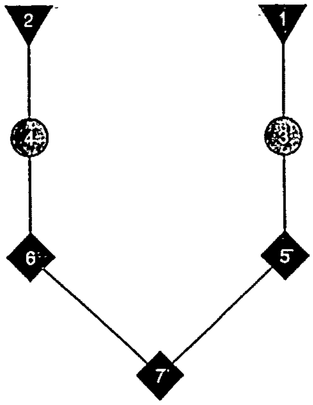


Figure 2: Best no-fill partition for a reordered graph. Only three factors are required.

Definition 1 *A partition (1) in which the factors P_k are invertible in place is called a no-fill partition. A no-fill partition of L with the smallest possible number of factors. is a* best no-fill partition.

Let $G(L)$ be the digraph with vertices $V = \{1, 2, \ldots, n\}$ and directed edges $E = E(L) \equiv \{(i,j) \mid i > j,\ L_{ij} \neq 0\}$. $G(L)$ is an acyclic digraph, or DAG. Consider the matrix L with graph $G(L)$ illustrated in Figure 1. L has a best no-fill partition:

$$L = (L_1)(L_2)(L_3)(L_4)(L_5)(L_6 L_7) .$$

This partition has six factors. It is possible to symmetrically permute the rows and columns of L such that L remains a lower triangular and $G(L)$ is as illustrated in Figure 2. A best no-fill partition of this reordered L is

$$L = (L_1 L_2)(L_3 L_4)(L_5 L_6 L_7) ,$$

which has only three factors.

*This work was supported in part under NSF Contracts ECS-8822654 and ECS-8907391.

†This work was supported by the NAS Systems Division under Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA).

Definition 2 *A* best reordered partition *of L is a symmetric permutation of the rows and columns of L such that the permuted matrix is triangular and has a best no-fill partition with the fewest possible factors.*

The aims of this work are as follows. We shall develop a theory of efficient algorithms for computing best no-fill (Section II) and best reordered (Section III) partitions. Second, we shall determine how useful these ideas are in practice by means of some experiments (Section IV).

II. BEST NO-FILL PARTITIONS

The *transitive closure* of a digraph $G = (V, E)$ is the graph $G\prime = (V, E\prime)$ where $E\prime = \{(i,j) \mid$ there is a $j \rightarrow i$ path in $G\}$. The digraph $G = (V, E)$ is *transitively closed* if it is equal to its own transitive closure.

We shall state our results and refer the reader to the full paper [1] for proofs.

Lemma 1 *If L is a nonsingular lower triangular matrix, then $G(L^{-1})$ is the transitive closure of $G(L)$.*

Let $G = (V, E)$ be a digraph associated with a triangular matrix L. Given a subset S of V, define the *column subgraph* $G_S = (V, E_S)$ (where $E_S \equiv \{(i,j) \in E | j \in S\}$) as the graph of the lower triangular matrix obtained by zeroing all columns of L not in S.

Theorem 2 *Let a partition (2) and corresponding factorization (1) be given. The factors P_k are invertible in place iff each column subgraph $G(P_k) = G_{\{e_{k-1}+1,\ldots,e_k\}}$ is transitively closed.*

Proof: By Lemma 1 $(P_k^{-1})_{ij} \neq 0$ iff there is a $j \rightarrow i$ path in $G(P_k)$. The following are therefore equivalent:

- P_k is invertible in place;
- $(P_k^{-1})_{ij} \neq 0 \Rightarrow (P_k)_{ij} \neq 0$;
- for every $j \rightarrow i$ path, $(P_k)_{ij} \neq 0$;
- $G(P_k)$ is transitively closed. □

The following algorithm was proposed by Alvarado, Yu and Betancourt [2]:

Algorithm P1:
Input: $L = L_1 L_2 \cdots L_{n-1}$
Output: A best no-fill partition of L.

```
$i \leftarrow 1$;   $k \leftarrow 1$;
while ($i < n-1$) do
    let $r$ be the largest integer greater than $i$ such that $L_1 \cdots L_r$
            is invertible in place;
    $P_k \leftarrow L_i \cdots L_r$;
    $k \leftarrow k+1$;   $i \leftarrow r+1$;
od
```

We have shown [1] that Algorithm P1 determines a best no-fill partition. (There may be others. Best no-fill partitions are not unique.)

III. BEST REORDERED PARTITIONS

This section describes a straightforward "greedy" algorithm for finding best reordered partitions.

For $(i,j) \in E$ we say that j is a *predecessor* of i and i is a *successor* of j. Let $G = (V, F)$ be a DAG. We define level(i), $i \in V$ to be the length of the longest path in G ending at i.

The algorithm works by finding a partition $V = \cup_{k=1}^{m} S_k$ for which the column subgraphs G_{S_k} are transitively closed. Moreover, S_1 is a source node in the quotient graph, i.e. there are no edges directed into S_1. If S_1 and its out edges are removed, then S_2 is a source node, etc. We shall call the subsets S_k in this partition *factors* in analogy with the corresponding factors P_k of L.

Algorithm RP1 (Re-order, Permute 1):
Input: A directed, acyclic digraph $G(L)$.
Output: A permutation $\nu : V \longrightarrow \{1, \ldots, n\}$ and a partition of L.

```
Compute level($v$) for all $v \in V$;
max-level $\leftarrow \max_{v \in V}$(level($v$));
$i \leftarrow 1$;   $k \leftarrow 1$;   $e_0 \leftarrow 0$;
while $i < n$ do
  $S_k \leftarrow \emptyset$;   $e_k \leftarrow i$;
  $\ell \leftarrow min\{j \mid$ there is an unnumbered vertex at level $j\}$;
  repeat
    for every vertex $v$ at level $\ell$ do
      if ((([Condition 1] $v$ is unnumbered ) and
         ([Condition 2] Every predecessor of $v$ has been numbered ) and
         ([Condition 3] Every successor of $v$ is a successor of all
              $u \in S_k$ such that $u$ is a predecessor of $v$) ) then
         $\nu(v) \leftarrow i$;   $i \leftarrow i+1$;
         $S_k \leftarrow S_k \cup \{v\}$;   $e_k \leftarrow e_k + 1$;
         fi
      od
    $\ell \leftarrow \ell + 1$;
    until $\ell >$ max-level or no vertices at level $\ell - 1$ are in $S_k$;
  $P_k \leftarrow L_{e_{k-1}} \cdots L_{e_k}$;   $k \leftarrow k+1$;
  od
```

Theorem 3 *Procedure RP1 finds a best reordered partition of of L.*

The complexity of Algorithm RP1 can be large. Consider a dense lower triangular matrix of order n. RP1 takes $O(n^3)$ time in this case, since the cost of checking whether all successors of vertex j are also successors of its predecessors is $O(j(n-j))$. In [1] we refined Algorithm RP1, producing an improved algorithm for which we can prove an $O($ nonzeros$(L))$ complexity bound. Furthermore, A. Pothen has developed a method with an $O(n)$ complexity bound for the case where the undirected graph $G(L + L^T)$ is chordal (this happens when L is a Cholesky factor) [6].

While our work is related to the use of clique and clique tree representations of sparse matrix factors [7], the partitioning of this paper is not the same as the partitioning into simplicial cliques or supernodes that has appeared previously. Indeed, all members of a simplicial clique of $G(L)$ are included in the same factor by Algorithm RP1, but several simplicial cliques may belong to the same factor; see [1].

IV. EXAMPLES

This section illustrates the performance of the proposed algorithms. Several examples compare P1 and RP1 with respect to the number of factors in the partitions they find. We also examine the effect of the initial ordering of rows and columns of a matrix A on the number of factors in a best reordered partition of its Cholesky or incomplete Cholesky factor L. An experiment shows that on the Connection Machine, the solution process (3) can be faster than substitution by orders of magnitude.

First, we compare algorithms P1 and RP1. Table 1 uses five power system matrices ranging is size from 118 to 1993. Table 2 gives results for matrices arising from five-point finite difference discretizations. In each case, the original coefficient matrix is first ordered and its Cholesky factor L is found. We need to distinguish this first fill-reducing ordering of A from the reordering of L found by RP1. We call the ordering of A the primary ordering. Three primary ordering procedures are used: the minimum degree algorithm [8], the multiple minimum degree (MMD) algorithm [5], and the minimum level, minimum degree (MLMD) algorithm [3].

For each matrix and primary ordering algorithm, two partitioning methods are compared: Algorithm P1, which simply partitions L optimally without reordering it, and Algorithm RP1 which reorders the matrix and generates an optimal partition. In most cases, Algorithm

	Min. Degree		MMD		MLMD	
Size	P1	RP1	P1	RP1	P1	RP1
118	53	14	10	10	6	5
352	132	21	13	12	8	8
707	213	26	23	18	11	10
1084	309	26	33	24	14	11
1993	563	35	41	25	15	15

Table 1: Effect of primary ordering on number of factors in best no fill partitions (P1) and best reordered partitions (RP1) for power system matrices.

	Min. Degree		MMD		MLMD	
Grid Size	P1	RP1	P1	RP1	P1	RP1
5-by-5	10	7	6	6	5	5
10 by 10	20	12	15	11	9	7
15 by 15	20	12	16	14	11	8

Table 2: Effect of primary ordering on number of factors in best no fill partitions (P1) and best reordered partitions (RP1) for 5-point difference operators on grid graphs.

RP1 gives a smaller number of factors than PA1, while in a few cases both algorithms give the same number of factors.

We observe that RP1 reduces the number of factors at no expense in added fills. Its effect is most dramatic if the underlying primary ordering is the minimum degree algorithm. On the other hand, the best results are obtained when the MLMD algorithm is used for the primary ordering, even though the relative improvement attainable by Algorithm RP1 over Algorithm P1 is small. The results for the MMD algorithm fall somewhere between minimum degree and MLMD. The reduction in the number of factors achieved by RP1 in comparison with P1 is quite dramatic when MMD is the primary ordering.

The second experiment we report compares the solution procedure (3) with the usual forward substitution method on the Connection Machine model CM-2, a highly parallel SIMD computer. We begin with a large sparse matrix A of order 4037, obtained from a triangular mesh in the region around a three-element airfoil. Three matrices L_1, L_2, and L_3 are obtained by approximate factorization.

L_1 is obtained by an incomplete LU factorization of A; we carry out the Gaussian elimination process, but we allow nonzeros in L (and U) only where there is a nonzero in A^2. The ordering of A is obtained from a lexicographic sort of the (x, y) coordinates of the grid which leads to the matrix; this ordering produces a large number of levels in $G(L)$.

L_2 is the incomplete LU factor obtained when a variant of MLMD is used as the primary ordering of A.

L_3 is the exact lower triangular factor of A, with the same primary ordering as foi L_2.

In Table 3 we give the size of these factors, the number of levels, which is proportional to the time requred for our parallel substitution algorithm, and the number of partitions, which is in practice proportional to the time required by the partitioned solution algorithm (3).

The computations used 8192 processors on the Connection Machine the NAS Systems Division, NASA Ames. From these results, as well as those above, we see that unless L has a fairly rich structure there is no great advantage to the use of the partitioned method. The use of an MLMD primary ordering improves both substitution and partitioned methods. However, with the introduction of the additional fill in the exact factor L_3 (compared with L_2), the number of levels in $G(L)$ increases sharply (as does the time for substitution) while the number of factors in the best reordered partition drops dramatically. The difference in the solution time, even for this problem of modest size, is about a factor of twenty. Thus, we conclude that the method can be quite useful in highly parallel machines when the matrix L has a rich enough structure, as happens when it is an exact triangular factor.

V. CONCLUSIONS

An algorithm for best reordered partitioning of lower triangular matrices has been presented and proven to be optimal.

Experiments have shown the method to be very valuable in practice on highly parallel machines.

For a lower triangular factor of a sparse matrix A, the number of partitions attainable is strongly influenced by both the ordering of rows and columns of A and the method of computing L.

References

[1] F. L. Alvarado and R. Schreiber. Optimal parallel solution of sparse triangular systems. RIACS Technical Report 90.36, September, 1990. Submitted to *SIAM J. Sci. Stat. Computing.*

[2] F. L. Alvarado, D. Yu and R. Betancourt. Partitioned sparse A^{-1} methods. *IEEE Transactions on Power Systems* 5, (1990), pp. 452–459.

[3] R. Betancourt. An efficient heuristic ordering algorithm for partial matrix refactorization. *IEEE Transactions on Power Systems* 3 (1988), pp. 1181–1187.

[4] S. W. Hammond and R. Schreiber. Efficient ICCG on a shared memory multiprocessor. Technical Report 89.24, Research Institute for Advanced Computer Science, 1989.

[5] J.W. Liu. Modification of the minimum degree algorithm by multiple elimination. *ACM Transactions on Mathematical Software* 11 (1985), pp. 141–153.

[6] A. Pothen. Fast parallel solution of a triangular system in sparse Cholesky factorization. In preparation.

[7] A. Pothen and C. Sun, Compact clique tree data structures in sparse matrix formulations. Computer Sciences Technical Report Number 897, December 1989, The University of Wisconsin, Madison.

[8] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proc. IEEE* 55 (1967), pp. 1801-1809.

Matrix	Ordering	Factor-ization	nonzeros	Levels in $G(L)$	Substitution Time	Factors	Partitioned soln. time
L_1	RCM	ILU	23,526	823	16.17 secs	816	15.73 secs
L_2	MLMD	ILU	26,793	78	2.20 secs	66	1.84 secs
L_3	MLMD	exact	118,504	311	28.89 secs	16	1.51 secs

Table 3. Comparison of CM-2 execution times for substitution and partitioned solution.

SUPERCOMPUTERS: THE HARDWARE, THE ARCHITECTURE

Willi Schönauer and Harmut Häfner
Rechenzentrum der Universität Karlsruhe
Postfach 6980, D-7500 Karlsruhe 1, Germany

Abstract: The prototype supercomputer and some leading architectures including massively parallel computers are discussed. A performance formula explains the gap between theoretical peak and real performance. The price/performance relation for the discussed computers is presented.

1. THE PROTOTYPE VECTOR COMPUTER

In Fig 1 the 'prototype' vector computer is depicted The technology is characterized by the cycle time in nsec. The floatingpoint units are pipelines, thus have a startup time and need long vectors for efficient use. Chaining means coupling and overlapped operation of different pipelines, e.g. load, multiply, add, store. There may be internal parallelism by multi-track pipelines, e.g. 4-track pipelines deliver 4 results per cycle.

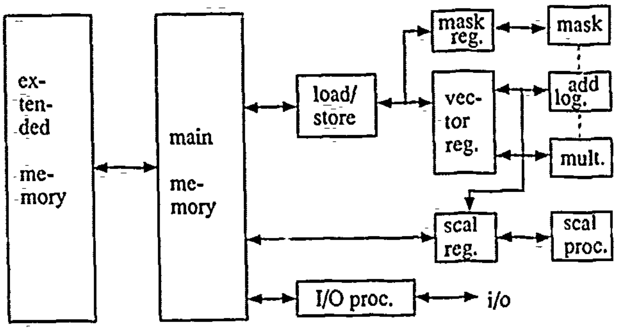


Fig: 1 'Prototype' vector computer.

In this paper we denote by 'word' 64 bits = 8 bytes and by 'pipe group' an addition and multiplication pipe. Critical points of vector computers are: memory bandwidth in words per cycle and pipe group; memory size in Mwords (million words); size and bandwidth of extended memory (the two sizes limit the problem size); i/o bottleneck: disks are extremely slow compared to the pipeline speed.

2. ARITHMETIC OPERATIONS AND MEMORY BANDWIDTH

Dyadic operations like

$$c_i = a_i + b_i \qquad (1)$$

need 2 loads and 1 store per cycle and pipe group. Triadic operations allow parallel operation of the addition and multiplication pipeline and deliver two results per cycle and pipe group, called supervector speed. The vector (or full) triad

$$d_i = a_i + b_i * c_i \qquad (2)$$

needs 3 loads and 1 store per cycle and pipe group. This is the most important operation and thus our key operation. More special is the linked triad with one scalar operand

$$c_i = a_i + s * bi \qquad (3)$$

with two loads and one store and still more special is the (repeated) contracting linked triad

$$b_i = b_i + s * a_i \qquad (4)$$

that needs only one load if b is fixed in a vector register. This is the basic operation of matrix multiplication. A vector computer with a memory bandwidth of one word per cycle and pipe group fits only to (4) and delivers only 1/4 of the peak performance for (2).

Because of the different number of memory references one should not count for vector computers merely additions and multiplications but count operations like e.g. linked triads.

3. SOME VECTOR COMPUTER ARCHITECTURES

In [1] detailed discussions of the most relevant supercomputers, including kernel program measurements, are presented. Present supercomputers are the continuation of these architectures. In the oral presentation overviews like Fig. 1 are presented for the different supercomputers. Here we present only their main characteristics for the maximal configurations. We discuss 3 real supercomputers, an integrated vector processor, a mini-supercomputer and a super-workstation.

CRAY Y-MP8/8256, cycle time 6 nsec, theoretical peak performance 2.67 GFLOPS, 8 processors (MIMD shared memory computer), 256 Mwords MM (main memory). 256 banks, 5 cycles bbt (bank busy time), bandwidth. 2 load and 1 store per cycle and pipe group, 2 Gwords EM (extended memory): bandwidth 2 * 1.3 GB/sec.

Fujitsu VP2600 (Siemens S600), cycle time 3.2 nsec, theoretical peak performance 5.0 GFLOPS, monoprocessor, but internally 2 4-track pipe groups, 256 Mwords MM. 512 banks, 14 cycles bbt, bandwidth. one word per cycle and pipe group, 1 Gword EM. bandwidth 2 GB/sec. Model VP2600/20 with two scalar processors, model VP2400/40 with two vector (2 2-track pipe groups) and 4 scalar processors.

NEC SX-3, Model 44, cycle time 2.9 nsec, theoretical peak performance 22 GFLOPS, 4 processors (MIMD shared memory), 8-track pipe group, 256 Mwords MM: 1024 banks, 7 cycles bbt, bandwidth: 0.5 words per cycle and pipe group globally (between memory and memory access unit), but 1 load and 0.5 store per cycle and pipe group locally (between memory access unit and processor), 2 Gwords EM: bandwidth 2.75 GB/sec. One or two control processors for operating system, with separate memory. Remark. Only the two processor Model 24 can be recommended (11 GFLOPS) if at least 1 word per cycle and pipe group global memory bandwidth is requested.

IBM ES 9000/720 with 6 VFs (Vector Facilities = integrated vector processors), corresponds to former 3090/600J, basically general purpose computer, cycle time 14.5 nsec, theoretical peak performance 828 MFLOPS, 6 processors (MIMD shared memory), 64 Mwords MM: transparent (no banking), bandwidth: one word per cycle and pipe group, 512 Mwords EM: bandwidth 2 * 138 MB/sec. Cache of 256 KB, danger of cache stumbling for long vectors. Announced: ES 9000/900, cycle time 9.5 nsec, 2526 MFLOPS, 6 processors, improved cache, two-track pipes.

CONVEX C240, Minisupercomputer, cycle time 40 nsec, theoretical peak performance 200 MFLOPS, 4 processors (MIMD shared memory), 256 Mwords MM: 128 banks, 8 cycles bbt, bandwidth: one word per cycle and pipe group, no EM. Soon expected: C300, cycle time 16 nsec, 8 processors, 1 GFLOPS theoretical peak performance, 512 Mwords MM.

IBM RISC SYSTEM 6000, Model 550, Super-Workstation, cycle time 24.4 nsec, theoretical peak performance 82 MFLOPS (can 'simulate' vector operations), 64 Mwords MM, transparent, bandwidth: one word per cycle.

4. PARALLEL COMPUTERS

We denote by 'parallel computer' an architecture that can be extended to 'many' processors. With parallel computers the real problems are shifted from the hardware level to the software level, i.e. to the user. We discuss only MIMD-type parallel computers. There are three basic types, see Fig. 2.

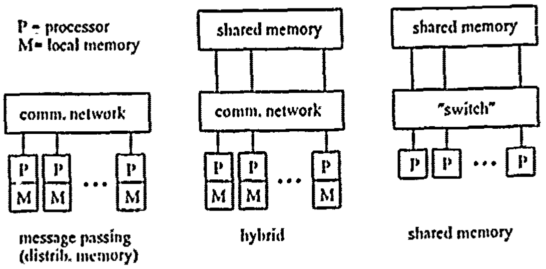


Fig. 2 Three basic type parallel architectures.

The [illegible] of shared memory computers is the memory bottleneck that limits the number of processors, and the memory contention. The user has to distribute only the processing.

The problem of the message passing system is the distribution of the data to the local memories, and the resulting communication overhead. Idling processors of a dedicated (sub)system cannot be used by other jobs, thus we have a GFLOPS PC The dream of the message passing community is the virtual shared memory, but this is just contrary to the communication efficiency.

Hybrid systems are the worst of all worlds because they combine the disadvantages of both inherent systems.

An essential drawback of parallel computers is that efficient programs must be tailored to the special architecture. In the following we discuss briefly two representative hypercube architectures, in the oral presentation detailed overview graphs will be presented.

NCUBE2: max. 13-dimensional hypercube, 8 - 8192 scalar 64-bit processors, max. 20 GFLOPS for 64 bit. The basic processor is a 1.2 micron CMOS custom processor with 0.5 M transistors, 20 MHz, 50 nsec cycle time, delivering 2.4 MFLOPS (the pipeline formula is not applicable), with 14 bidirectional DMA channels with 2.2 MB/sec, each (one is for i/o). One node has 1 or 4 or 16 MB local memory. Subcubes can be allocated to different users (space sharing. 'PCs'). The job management, the compilers and the tools are running on the SUN workstation that serves as host computer.

INTEL iPSC/860: max. 7-dimensional hypercube, 8 - 128 i860 vector processors, max. 5.1 GFLOPS for 64 bit. The basic i860 processcr is a 0.5 micron CMOS custom processor with 1 M transistors, 40 MHz, 25 nsec cycle time. The pipelined floating point units deliver 40 MFLOPS 'supervector speed' because the multiply unit needs 2 cycles for one result. Unfortunately presently the software is far behind the hardware, and Fortran compilers are unable to reach this performance. There is a one-word (64 bits) per two cycles memory bottleneck between i860 chip and memory. The node has 8 or 16 MB local memory and 8 bidirectional communication links with 2.8 MB/sec, each (one is for i/o). Subcubes can be allocated to different users (space sharing: 'PCs'). The job mangement, the compilers and the tools are running on an INTEL 80386 PC that serves as host computer.

5. PERFORMANCE FORMULA

In this section we want to explain why there is such a large gap between theoretical peak performance and real performance for present supercomputer architectures. If we take 1000/τ , where we measure the cycle time τ in nsec, we get the (vector) speed of a single pipeline in MFLOPS, times two yields the supervector speed, times the number P of total pipe groups in the system yields the theoretical peak performance. But unfortunately the latter is reduced by several reduction factors f_i that we want to discuss below. Thus our formula for the real performance of a pipelined supercomputer is

$$r_{real.} = \underbrace{\frac{1000}{\tau[nsec]} * 2 * P}_{theoret.\ peak} * \underbrace{f_1 * f_2 * . . . * f_N}_{reduction\ factors} \ [MFLOPS] \ . \qquad (5)$$

Repeatedly internally lost cycles (e.g. section loop organization): for m cycles we have in the mean m * d lost cycles, thus

$$f_1 = \frac{1}{1+d} , \qquad (6)$$

e.g. d = 0.05, f_1= 0.95.

Memory bottleneck for vector triad (2). only the vector triad has full flexibility of multiplication and addition as 'assumed' in the peak performace, all other triads are 'exceptions'. Therefore we select as key operation the vector triad that needs 4 memory references per cycle and pipe group. If we have instead of 4 only m, we get

$$f_2 = \frac{m}{4} , \qquad (7)$$

e.g. m = 1, f_2 = 0.25.

Finite vector length n (startup cycles lost once). If we denote by $n_{1/2}$ Hockney's half performance length, see [1,2], we get

$$f_3 = \frac{n}{n+n_{1/2}} , \qquad (8)$$

e.g. n = 1000, $n_{1/2}$ = 1000, f_3 = 0.91.

Scalar code. part v of operations is vectorizable, part (1 v) is scalar, w is the ratio vector/scalar speed for infinitely long vectors. Then we get

$$f_4 = \frac{1}{(1-v)w+v} . \qquad (9)$$

This is Amdahl's law for vectorization, see [1,2], e.g. w = 10, v = 0.95, f_4 = 0.69.

Up to now we have discussed monoprocessor vector computers. For the examplary values we get f_1 * f_2 * f_3 * f_4 = 0.15, i.e. our supercomputer would deliver only 15 % of its theoretical peak performance for the vector triad.

In the following we discuss parallel computers with p processors.

Sequential (non-parallelizable) code. part q of operations is parallelizable, part (1-q) is sequential (fine grain parallelism). Then we get

$$f_5 = \frac{1}{(1-q)p+q} . \qquad (10)$$

This is Amdahl's law for parallelization, compare to (9), e.g. p = 64, q = 0.95, f_5 = 0.24.

Shared memory MIMD parallel computer. memory contention. If a denotes the ratio of available over minimal number of memory banks and c denotes the part of operations with contiguous, (1-c) with random elements (indirect addressing), then a model yields

$$f_{6,sh} = \frac{1}{1+2(1-c)/a} , \qquad (11)$$

e.g. a = 2, c = 0.95, $f_{6,sh}$ = 0.95.

Message passing (distributed memory) MIMD parallel computer. waiting for communication. For m useful cycles in the mean m * b additional cycles are needed for non-overlapping communication. Then we get

$$f_{6,mp} = \frac{1}{1+b} \qquad (12)$$

e.g. b = 0.05, $f_{6,mp}$ = 0.95.

If we calculate up to this point f_1 * ... * f_6 = 0.034, we see that we get with those 'reasonable' assumptions finally not more than 3.4 % of the peak performance, i.e. we have lost 96.6 % of our supercomputer'!

Load balancing (coarse grain parallelism). we assume that a user has reserved p processors (e.g. his subcube), t_i is the time that processor i is active. Then we get his 'personal' utilization factor

$$f_7 = \left(\sum_{i=1}^{p} t_i \right) \ / \ (p * \max_i t_i), \qquad (13)$$

e.g. f_7 = 0.9.

Long-range continuous usage: an idling computer produces no GFLOPS. Thus the global utilization factor

$$f_8 = \text{(hours of usage per year)}/8760 \qquad (14)$$

is of decisive importance. For a workstation that is used (as number cruncher) for 8 h on 5 days a week f_8 = 0.24.

6. PRICE/PERFORMANCE RELATIONS

The price/performance relation has two components. The price is determined by the selected configuration that is strongly determined by the size of the memory. We select a configuration for the solution of large problems, else a supercomputer is not an appropriate tool. Therefore we select (as far as possible) 1 GB MM (main memory), 1 GB EM (extended memory) and 100 GB disks. P denotes the purchase price, given in MDM (million deutschmark), M+L denotes maintenance and licence costs, given in MDM/a (per annum = year). All prices are commercial list prices (no university discount), without VAT, valid January 1991. The software for a FORTRAN environment with operating system, compiler and tools is included.

The performance is given by the peak performance and the reduction factors. We determine the performance for the vector triad. We note only the reduction factors ≠ 1 that are applied.

We attribute some personnel to the computer. SE denotes system engineer with 78 KDM/a, OP denotes operator with 52 KDM/a. We do not consider cost for housing, electricity, climatization.

The price/performance relation is given in MDM/a per GFLOPS, i.e. the cost to be paid per year to have 1 GFLOPS sustained vector triad. For this purpose the purchase price is distributed onto four years, i.e. we assume a four year life-cycle of our supercomputer. In the following we give the data for the different computers.

CRAY Y-MP8/8128: 1 GB MM, 1 GB EM, 100 GB disks, P = 48.05 MDM (31 M$), M+L = 2.16 MDM/a, 2 SE, 3 OP, peak performance 2.67 GFLOPS, f_2 = 3/4, $f_{6,sh}$ = 0.9 (estimated).

Fujitsu (Siemens VP2600/10, 1 GB MM, 1 GB EM, 100 GB disks, P = 28.65 MDM, M+L = 1.48 MDM/a, 2 SE, 3 OP, peak performance 5.0 GFLOPS, f_2 = 1/4.

NEC SX-3, Model 24: 1 GB MM, 1 GB EM, 100 GB disks, P = 30.48 MDM, M+L = 1.37 MDM/a, 2 SE, 3 OP, peak performance 11.03 GFLOPS, $f_2 = 1/4$, $f_{6,sh} = 0.95$ (estimated).
IBM ES9000/720 (3090J/600), 6VFs: 0 5 GB MM (max), 1 5 GB EM, 100 GB disks, P = 38.24 MDM, M+L = 1.829 MDM/a, 2 SE, 3 OP, peak performance 0.828 GFLOPS, $f_2 = 1/4$, $f_{cache\ stumbling} = 0.8$ (estimated).
CONVEX C240: 1 GB MM, 8 GB disks, P = 3.9 MDM, M+L = 0.39 MDM/a, 1 SE, peak performance 0.2 GFLOPS, $f_2 = 1/4$, $f_{6,sh} = 0.9$ (estimated). (Remark: 1 GB MM is unusually large for a CONVEX and brings the price up, but for the solution of large problems we need a large MM.)
IBM RISC SYSTEM/6000, Model 550 Workstation: 0.5 GB MM (max.), 2.5 GB disks, P = 953 KDM, M+L = 26.5 KDM/a, 0.1 SE, peak performance 0.081 GFLOPS, $f_2 = 1/4$, $f_8 = 0.24$ or 0.75. (Remark: The maximal MM of 0.5 GB excludes this workstation from the solution of very large problems. The price for the memory brings the price for the workstation up.)
NCUBE 2, Model 10: 512 processors à 4 MB → 2 GB MM, 50 GB disks, P = 7.0 MDM, M+L = 0.70 MDM/a, 2 SE, peak performance 1.23 GFLOPS, f_2 = 1 (scalar!), $f_5 = 0.95$ ($q = 0.9999$), $f_8 = 0.1$ or 0.24 or 0.75. (Remark. The 2.4 MFLOPS per processor are based on parallel scalar execution of addition and multiplication and are met for the vector triad by the 80 MB/sec between CPU and its local memory.)
INTEL iPSC/860, Model 128: 128 processors à 16 MB → 2 GB MM, 50 GB disks, P = 9.8 MDM, M+L = 0.98 MDM/a, 2 SE, peak performance 5.1 GFLOPS, $f_2 = 1/4.5$, $f_5 = 0.99$ ($q = 0.9999$), $f_8 = 0.1$ or 0.24 or 0.75. (Remark. The value $f_2 = 1/4.5$ results from the fact that a load for 64 bits needs two cycles, but an immediately following store needs 3 cycles, thus 9 = 4.5 * 2 cycles are needed for the vector triad and 40/4.5 = 8.9 MFLOPS per i860 chip result. An assembler program should come close to this value, a vectorizing compiler in a test has obtained 7 MFLOPS according to INTEL. But the software that has been delivered up to now is far behind the hardware possibilities.)

Everybody is free to choose his own parameters in this play. In Fig. 3 the price/performance relation for our parameters is depicted. The results speak for themselves. The general purpose computer has its own merits, but it is an expensive number cruncher. Nevertheless the purchase of VFs pays if the ES9000 is used in scientific computations. The mini-supercomputer is 'relatively' expensive in spite of the 'cheap' technology because of the 'relatively' large memory. For the workstation and the parallel computers the utilization factor is decisive for the price/performance relation. These computers have (not yet) a batch environment that allows the continuous usage of the computer like for a conventional computer.

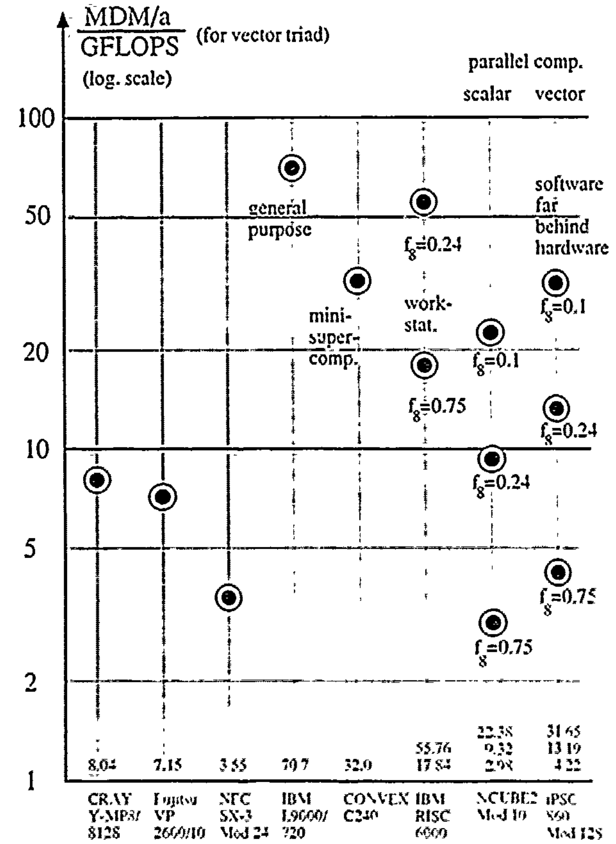


Fig. 3 Price/performance relation for the selected parameters of the different computers.

We have presented list-prices. A discount may change the relations correspondingly. Note that the software results in additional losses that may increase the price/performance relation considerably. But finally an efficient use of any type of supercomputer is possible only with data structures that are tailored to its architecture. If the data structure does not allow sufficient vectorization and/or parallelization, Amdahl's law (9) and/or (10) will destroy any efficiency.

7. CONCLUDING REMARKS

The cycle time for the ECL-technology of the large supercomputers will continue to decrease from now 2.9 nsec (NEC SX-3) to perhaps 2 nsec (CRAY-3 ?) and eventually 1 nsec for 1995++ and 0.5 nsec in the year 2000. The CMOS-Technology of presently 40 MHz/25 nsec (i860 chip) will evolve to 50 - 80 MHz/20-12.5 nsec (12.5 nsec was the cycle time of the CRAY-1), INTEL expects for the year 2000 200 MHz/5 nsec. Thus the usual factor of 10 in speed between ECL and CMOS may be maintained.

The engineers want (sustained) performances of 100 GFLOPS, then 1000 GFLOPS = 1 TFLOPS. This can be obtained only by parallelism. But there are two main problems:

Problem 1. Can we pay the memories that are needed to store the operands in order to use TFLOPS? The answer is. We have to wait until we can afford these memories, increased parallelism does not solve this problem.

Problem 2. How can we organize a user-friendly parallelism on the hardware level? The answer is presented in [3].

REFERENCES

1. W. Schönauer, Scientific Computing on Vector Computers, North-Holland, Amsterdam 1987.

2. R.W. Hockney, C.R. Jesshope, Parallel Computers 2, Adam Hilger, Bristol 1988.

3. W. Schönauer, R. Strebler, Could user-friendly supercomputers be designed? in J.T. Devreese, P.E. van Camp (Eds.), Scientific Computing on Supercomputers II, Plenum Press, New York 1990, pp. 99 - 122.

Supercomputers: Numerical Libraries

Nikolaus Geers
University of Karlsruhe, Computing Center
Postfach 6980
D-7500 Karlsruhe 1

Abstract

Different numerical libraries from hardware manufacturers and software houses are available for supercomputers. Contents and performance of these libraries on CRAY 2, CRAY Y-MP, Fujitsu/Siemens VP-series and IBM 3090 VF is discussed and a proposal for future library development is given.

1 Introduction

Numerical subprogram libraries are important tools in developing scientific software. With the increasing use of supercomputers, in most cases pipelined vector computers, there is an increasing need for reliable and efficient collections of numerical subprograms. This frees the application programmer from recoding numerical algorithms and increases the reliability and maintainability of application programs.

This paper will concentrate on discussion of efficiency of libraries on supercomputers but will not include a general evaluation of different libraries since this must comprise a detailed discussion of the numerical algorithms, their robustness and their implementation which is beyond the scope of this paper.

In section 2 a brief review of different libraries for supercomputers will be given. Since most work on adapting libraries to supercomputer architectures has been done in the area of linear algebra this is discussed in greater detail in section 3. In the concluding remarks some items missing in todays subroutine libararies are listed and a proposal for future library developments on supercomputers is given.

2 Numerical Libraries for Supercomputers

Two groups of numerical subroutine libraries for supercomputers must be distinguished:

- Libraries developed by hardware manufacturers and being a part of the software environment on a given computer system,
- Libraries developed by software vendors (e.g. IMSL and NAG) which are in general available on a wide range of different computer systems.

2.1 Libraries from Computer Manufacturers

Cray: SCILIB The Scientific Library (SCILIB), release 5.0, contains programs from three different areas: linear algebra, Fast Fourier Transforms (FFT) and searching and sorting. The linear algebra chapter comprises all level 1, level 2 and level 3 BLAS as described in [6, 1, 2] except those level 2 BLAS for packed matrices. Also optimized versions of LINPACK and EISPACK are included in SCILIB. There are routines for single and multiple FFT's, the routines doing multiple transforms are of mixed radix type allowing radices 2, 3, 5 and 7. A random number generator is included in the runtime library of the compiler.

IBM: ESSL The Engineering and Scientific Subroutine Library (ESSL) from IBM is available in a vector as well as in a scalar version running on any IBM mainframe. In addition to the chapters on linear algebra, FFT, sorting and searching some other routines for interpolation, numerical quadrature, random number generation and parallel processing are included in this library. Only a subset of level 2 and level 3 BLAS are available. But the linear algebra chapter contains some routines for solving sparse linear systems by direct or iterative methods.

Fujitsu/Siemens. SSL II and supplemenents Fujitsu's SSL II library is supplemented by Siemens with a complete set of all three levels of BLAS, a set of FFT routines and random number generators. Together with these supplements SSL II seems to be the most comprehensive library from a supercomputer manufacturer. The linear algebra chapter contains several routines for solving linear systems and eigenvalue problems. There are routines for nonlinear equations, minimization, interpolation and approximation, Fourier and Laplace transforms, differentiation and quadrature, ordinary differential equations, special function approximation and random number generation.

2.2 Standard Numerical Libraries

While the manufacturer supplied subprogram libraries are available only on specific computer systems, standard mathematical subprogram libraries like IMSL/MATH library from IMSL Inc. and NAG Fortran library from the Numerical Algorithms Group Ltd. are available on a wide range of computer systems ranging from PC's to supercomputers. So program development and small production runs can be done on a workstation while large production runs are executed on a supercomputer. The identical program can be executed in different supercomputer environments.

Versions of these libraries are available for the most important supercomputers like CRAY Y/MP, CRAY 2, IBM 3090 VF, NEC SX-series or Fujitsu/Siemens VP-series.

These libraries have not been designed for use on supercomputers from their starting point. But during the last years IMSL as well as NAG have spent much work in adapting the libraries to achieve efficient implementations on supercomputers. This must been done while maintaining the portability of the library, i.e.

- The user interface of library routines must not be changed. But the user interface of new routines which are introduced into the library may be choosen to allow efficient vectorization.
- The source code of library routines should be identical on all computer systems as far as possible. This is necessary

in order to guarantee the quality of the software and to maintain the software over a long life cycle.

Both libraries, IMSL as well as NAG, have a much broader contents than the libraries from supercomputer manufacturers. They comprise programs for solving systems of linear or nonlinear equations, eigensystem analysis, interpolation and approximation, integration and differentiation, differential and integral equations, Fourier and Laplace transforms, linear and nonlinear optimization, special function approximation and a lot of utility functions. Besides this many statistical capabilities for data analysis are included. The routines from linear algebra chapters in most cases handle dense matrices, and a few routines in NAG library are available for sparse matrices, and IMSL library contains two routines for iterative solution of linear systems which operate on sparse matrices.

3 Adapting Libraries for Supercomputers

Work on adapting standard numerical libraries to the architecture of supercomputers must concentrate on those areas where most computing time is spent. These are linear algebra, FFT and solution of differential equations. An overview of vectorizing NAG and IMSL library is given in [3] resp. [7].

Besides restucturing of existing library routines new programs have been introduced into the libraries, especially into the NAG library, which are well suited for vectorization. So new routines for numerical quadrature and solution of differential equations have been incorporated which evaluate function values at many grid points in one subroutine call. Calling an external subprogram to evaluate only one function value would disturbe vectorization and would increase the CPU time significantly. Also new routines for FFT, doing several transforms in parallel have been included. This gives in general greater vector lengths and higher speed ups on vector computers.

In order to improve the efficiency of library routines on supercomputers a hierarchical programming concept must be applied. It is necessary to identify the most time consuming parts of programs which are general enough to be put into a library subset that can be modified and tuned for a specific supercomputer.

3.1 Linear Algebra

Basic Linear Algebra Subprograms (BLAS) have been defined in [6, 1, 2] and are widely recognized as a standard. Level 2 BLAS contain routines for matrix-vector-multiplication of rectangular, triangular, symmetric or hermitian matrices, rank-one- and rank-two updates of rectangular, hermitian or symmetric matrices and solution of linear systems with triangular coefficient matrix. Matrices may be stored in different storage modes (general, banded or packed). Level 3 BLAS contain similar routines for matrix-matrix-operations: matrix-matrix-multiplication of rectangular, symmetric, hermitian or triangular matrices, rank-k- and rank-2k-updates of hermitian or symmetric matrices and solution of linear systems with triangular coefficient matrices and multiple right hand sides. Carefully optimized versions of these routines are being supplied by most supercomputer manufacturers. Usage of these optimized BLAS will result in portable and very efficient software.

In the current version of IMSL- and NAG-library linear algebra routines are based on level 2 BLAS and a few routines already use level 3 BLAS. In the next releases level 3 BLAS will be used more intensively.

Table 1 shows the performance of some library routines for solving a system of linear equation based on LU factorization. L2TRG/LFSRG from IMSL, F04AAF from NAG, SGEFA/SGESL from SCILIB, DLAX from SSL II and DGEF/DSEL from ESSL. In all cases the array containing the matrix has been defined with an odd leading dimension in order to minimize memory bank conflicts.

Computer	Library	matrix size				
		100	200	300	400	500
Siemens S400/10	SSL II	155	322	439	518	577
	NAG	147	395	638	816	972
	IMSL	120	354	582	762	918
CRAY Y-MP	SCILIB	100	128	137	145	145
	NAG	104	168	202	223	238
	IMSL	114	197	238	258	267
IBM 3090 S VF	ESSL	46	65	75	80	83
	NAG	29	50	61	67	72
	IMSL	34	51	60	63	65
CRAY 2	SCILIB	134	214	228	253	284
	NAG	35	34	41	75	80
	IMSL	24	31	54	40	67

Table 1. Performance of LU-factorization in MFLOP/s

Table 2 shows similar results for Cholesky-factorization, using LFIDS/LFSDS from IMSL, F03AEF/F04AGF from NAG, SPOFA/SPOSL from SCILIB, DVLSX from SSL II and DPPF/DPPS from ESSL. Again the leading dimension of the array is odd except DVSLX and DPPF/DPPS which accept the matrix in packed storage mode in a one-dimensional array.

Computer	Library	matrix size				
		100	200	300	400	500
Siemens S 400/10	SSL II	144	316	434	518	582
	NAG	96	240	366	458	538
	IMSL	99	308	507	580	672
CRAY Y-MP	SCILIB	51	80	85	108	108
	NAG	119	191	226	248	261
	IMSL	119	208	246	267	277
IBM 3090 S VF	ESSL	48	71	80	86	89
	NAG	40	63	60	59	59
	IMSL	26	47	58	60	57
CRAY 2	SCILIB	26	35	38	41	42
	NAG	44	55	67	114	126
	IMSL	40	80	63	62	72

Table 2· Performance of Cholesky-factorization in MFLOP/s

In order to study the behaviour of these routines in case of badly dimensioned arrays the same programs have been executed on Siemens S 400/10 and CRAY Y MP for matrices of size 256 and 512 with different leading dimensions. The results are given in table 3 and indicate that the right selection of leading dimensions in the calling program can increase the performance significantly.

Similar results as those reported in tables 1, 2 and 3 can be obtained for other linear algebra routines showing that the performance of routines from standard numerical libraries like IMSL and NAG

Computer	Library	LDA = N		LDA = N+1	
		256	512	256	512
LU-factorization					
Siemens	SSL II	59	64	393	578
S 400/10	NAG	259	465	539	978
	IMSL	256	491	487	926
CRAY Y-MP	SCILIB	136	153	139	148
	NAG	175	227	190	239
	IMSL	127	147	226	272
Cholesky-factorization					
Siemens	SSL II [1]	388	588	388	588
S 400/10	NAG	236	426	307	543
	IMSL	233	479	428	676
CRAY Y-MP	SCILIB	42	46	96	125
	NAG	210	258	216	262
	IMSL	203	255	235	279

[1] This routine uses a one-dimensional array to store the matrix.

Table 3: Effect of leading dimension

is in general as good or even better than the performance of routines from other libraries. In most cases these routines give good performance over a wider range of parameters (e.g. array dimensioning). But a prerequisite is the usage of carefully optimized BLAS. Only on IBM 3090 VF the routines from ESSL run about 20% faster than routines from other libraries.

3.2 Fast Fourier Transforms

A comparison of the performance of different library routines for FFT shows much greater differences than in the area of linear algebra [5]. Table 4 gives timings for one complex FFT of varying length on a Siemens S 400/10. Similar performance ratios can be found on other supercomputers.

N	FFTVPLIB DFTCB1	NAG C06FRF	IMSL DF2TCF
64	0.029	0.018	0.176
128	0.039	0.056	0.209
256	0.041	0.068	0.225
512	0.054	0.103	0.336
1024	0.092	0.171	0.531
2048	0.154	0.382	0.922
4096	0.322	0.740	1.698
8192	0.656	1.495	3.382
15625	1.468	1.602	6.062
16384	1.203	3.009	6.598
19683	2.148	2.121	9.966
32768	2.685	12.216	12.960
65536	5.456	24.585	25.874
131072	10.636	50.105	53.865
262144	21.746	101.230	108.094

Table 4. Timing of different FFT-routines on Siemens S400/10

4 Conclusion

Subroutine libraries from supercomputer manufacturers cover only a small range of applications, most attention has been given to linear algebra and FFT. But in the field of linear algebra standard numerical libraries give equivalent or even better performance. This demonstrates that well defined and optimized low level routines, in this case the BLAS, allow the development of very efficient and portable software. There is a need for comparable basic routines in other areas. In [4] a first idea for a set of Basic Operations for Fourier Transforms (BOFT) is given which hopefully will play a similar rule for applications of FFT. Also discussions on sparse level 2 BLAS defining matrix-vector-operations for sparse matrices are going on. As with the BLAS these low level routines should become part of the software environment of a supercomputer. This is the basis for development of numerical libraries and application programs.

Today there are some open problems concerning numerical libraries on supercomputers:

- The application programmer must very carefully select the appropriate routine. Sometimes there are several library routines doing similar operations but using different data structures which may influence the performance significantly.
- When passing multidimensional arrays the leading dimension should always be an odd number. But some library routines do not include this as a separate item in the parameter list.
- Within one library the performance of different programs may vary significantly.
- Some important application areas (e.g. solution of sparse linear systems or sparse eigenvalue problems) are not covered by library routines.

References

[1] J.J. Dongarra, J.J. Du Croz, S.J. Hammarling and R.J. Hanson: *An extended Set of Fortran Basic Linear Algebra Subprograms*, ACM Trans. Math. Software 14, pp 1-17, 1988

[2] J. Dongarra, J. Du Croz, I. Duff, S. Hammarling: *A Set of Level 3 Basic Linear Algebra Subprograms*, ACM Trans. Math. Software 16, pp1-17, 1990

[3] J.J. Du Croz: *Vectorization Review: Part 1 and Part 2*, NAG Newsletter 1/89 and 2/89, NAG Ltd., Oxford, 1989

[4] O. Haan and W. Wälde: *FFTVPLIB, a Collection of Fast Fourier Transforms for Vectorprocessors*; in H. Burkhart (Ed.): CONPAR 90-VAPP IV, Lecture Notes in Computer Science, vol.457, pp 447-457, Springer-Verlag Berlin, Heidelberg, New York,1990

[5] O. Haan and W. Wälde: *Fast Fourier Transform Libraries for Vectorcomputers*; Supercomputer 40, pp 42-49,1990

[6] C. Lawson, R. Hanson, D. Kincaid and F. Krogh: *Basic Linear Algebra Subprograms for FORTRAN Usage*, ACM Transactions on Mathematical Software 5 (1979), 308-323

[7] P. W. Smith, R. J. Hanson, J. Li, T. R. Leite: *Supercomputing at IMSL*, IMSL Vectorization Report, Update 9001V, IMSL Inc., Houston, 1990

Supercomputers and Distributed Applications: Status, Trends, Needs

Klaus F. Hanauer and Andreas Knocke
Rechenzentrum der Universität Karlsruhe
Postfach 6980, D-7500 Karlsruhe 1, Germany

Abstract: The supercomputer in combination with the graphics super-workstation opens a new simulation technique. The scientific visualization enables computational scientist and engineers to undertake "human-in-the-loop" problems - a class requiring visualization techniques as different from photorealistic computer graphics as interactive computing is from batch processing. The developer of the necessary software systems that provide true "distributed applications" has to divide the application program to the existing resources - supercomputer, network and graphics super-workstation.

1. Introduction

The new series of powerful supercomputers, a new class of powerful workstations and the high speed LAN's in combination lead to a new class of applications called "distributed applications". The "graphics super-workstations" are expected to play an increasingly important role in providing an enhanced environment for supercomputer users.

Their potential uses include:

1. Off-loading the supercomputer
   - service station for the supercomputer as front-end system (input queue - output queue - print queue management)
   - pre- and postprocessing of the input and output of supercomputer applications
   - distributed or shared processing.

2. Scientific visualization
   - understanding of results
   - communication of results.

3. Real-time interaction with the supercomputer
   - controlling of iterative computations
   - kill, suspend and restart of supercomputer jobs
   - exploration and development of new algorithms.

2. Status

The term "graphics super-workstation" is defined here to refer to a category of workstations introduced in 1988 which combine high quality graphics with very powerful computational capability. Typically, such workstations provide from 1/10 to 1/100 the floating point speed of the most powerful current supercomputers, they have large main memories (16 - 256 MBytes) and are capable of generating and manipulating realistic, three-dimensional graphic displays. In combination with high speed LAN's it is possible to exchange data very fast between the different computers in the LAN and the workstation memory. The workstation provides a very rapid movement of data between memory, disk storage, computational units and graphics hardware. The extensive system and application software provides users with a powerful and convenient working environment. In the following table typical graphics and display manipulation characteristics will be specified. In the second column of table 1 the actual capabilities of the workstation installed in the computing center of the University of Karlsruhe will be presented.

	typical graphics super workstation	workstation of the University of Karlsruhe
screen display	19" color monitor resolution 1280·1024	19" color monitor resolution 1280·1024
image bit-planes	16-32 bit Z-buffer 24 bit color planes provision for double buffering	2048·1024/plane 8 planes, 4 overlay planes 16 bit Z-buffer 16.7 Mio colors
display speed	500.000 3D vectors/sec 150.000 couraud-shaded, Z-buffered triangles/sec 30.000 Phong-shaded, Z-buffered triangles/sec	240.000 3D vectors/sec 50.000 triangles/sec without light-source 38.000 triangles/sec with light source
surface geometry approximation (primitives)	polygons, triangular strip meshes, NURBS	polygons, vectors, triangular strip meshes, NURBS (order of 6) dithering, back facing cull
shading, lightings, rendering	Flat-, Phong-, Gouraudshading, texturing, transparency, specular highlighting, ray tracing	Flat-, Phong-, Gouraudshading, radiosity, advanced hardware, lighting, transparency, ray tracing (optional)

Table 1: Typical graphics capabilities and values of the university workstations

Table 2 shows the typical values of a graphics super-workstation hardware and in the second column the corresponding values of the university machines.

The supercomputer performance, mainly the very fast floating point pipelines with a peak performance of 5 GFlops and the large main memory of 2 GByte is the reason that we need powerful peripheral workstations with super graphic capabilities to prepare the supercomputer results or to steer the simulation. Supercomputing is currently in the gigaworld era. Unfortunately, we may also be confronted with GBytes of output. This may occur not only from scientific problems that deal with very large amounts of input data, but also as output data from solutions to mathematical equations representing physical, chemical or technical processes.

At the third IFIP International Conference on Data Communication Systems and their Performance an empirical ratio of approximately 100 Bytes of output per MFlops of calculation was found. Hence, as we approach processing power of 5 GFlops of our Siemens S600/20 vector-computer, and a simulation time of 2000 seconds, this empirical ratio would predict the following output:

$$100 \text{ Bytes/Mflops} \bullet 5 \bullet 10^3 \text{ MFlops/sec} \bullet 2000 \text{ sec} =$$

$$1 \text{ GByte}$$

	typical graphics super-workstation	workstation of the University of Karlsruhe
number of processing units, floating point units	1-4 vector floating point pipes	1 central processing unit without vector capability
vector or floating point performance	5-30 MFlops	2 MFlops (BLAS)
integer processing	10-80 Mips	14 Mips
cache memory	1 MByte (1 GByte/sec)	128 KByte
main memory	16-128 MByte (300 MByte/sec)	16-24 MByte
I/O channels	80-100 MBit/sec	60 MBit/sec
disc storage	300-2000 MByte	300-1200 MByte

Table 2: Workstation hardware

In the computing center we can have up to 10 jobs per day of this 1/2 hour type, so we produce data output in an order of 10 GByte/day. These types of jobs are large scale calculations in the fields of finite element structural analysis, fluid dynamics involving repeated iterations over a spatial grid, and ab initio computational chemistry involving determination of eigenvalues of very large sparse matrices and multi-dimensional integrations.

The driving sources for increased processing speed, main memory and disc storage are the physical realism, the increased dimensionality and the data volume. Realistic representation of physical or technical systems may increase geometric complexity or eliminate simplifying approximations. On the other side, the dimensionality of a problem is not limited to the physical dimensions and the time, but more generally represents the number of degrees of freedom that must be considered as in the number of grid points in a computational fluid dynamics problem or the number of finite elements in a structural analysis problem.

Scientific problems of the departments of the University of Karlsruhe that are driving forces for the increase in computational power are.

- Computational Fluid Dynamics (including turbulence)
- Structural Analysis (finite elements, nonlinear analysis, different materials, eigenvalues, modification of the geometry)
- Physics/Chemistry (molecular dynamics, ab initio quantum chemistry, surface chemistry, statistical mechanics, astrophysics)
- Material Science (superconductivity, sinter materials, materials by design)
- Seismology
- Climate- and local weather simulation (environmental influences)

Structural analysis or computational fluid dynamics are typical engineering problems requiring a vector computer for the solution of the discretizised system using the finite element method employed in many commercially available codes as ADINA, FIDAP or LS-Dyna 3D. An example of graphical output is shown in figure 1. The 3D turbulent flow around a car has been computed on the Siemens/Fujitsu S600/20 supercomputer using FIDAP and the results are visualized on the workstation.

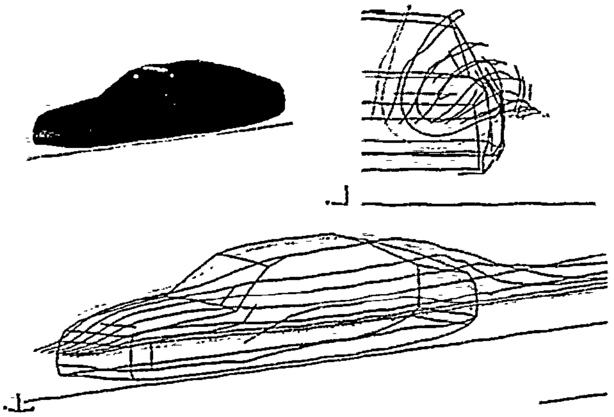


Figure 1 3D turbulent flow around a car

Generating the spatial discretization into different elements is done as preprocessing using a workstation freeing the supercomputer for the numerical intensive part of the analysis.

3. Graphics Super-Workstation in a Supercomputing Environment

The supercomputer of the University of Karlsruhe is needed to solve important scientific problems. The supercomputers of the future are needed to solve problems that presently cannot be done at all and should be designed and used for this purpose. The new generation of powerful workstations of the university provide a logical, cost-effective and user-time-effective alternative to shared supercomputers and indeed this is one of their appropriate and important roles in a supercomputing environment shown in figure 2. The increase in computational capability should not be the result of the aggregate demands of many users each of whom may need only a small amount of supercomputer time.

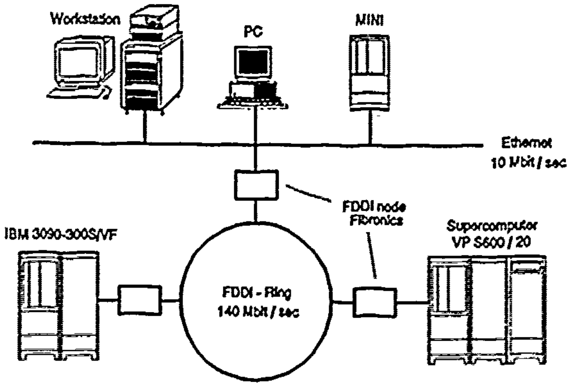


Figure 2 Supercomputing environment

Regarding the large amount of results and output data, the graphics super workstation can also play the role of a 'filter'. Only the necessary information to detect new phenomena arrives at the workstation screen. Along with processing speed and larger main memories, there is a need for increased storage and communication bandwidth. The present limitation of our LAN's of about 140 Mbit/sec is restricted but adequate for now, however the existing WAN's (excluding the BELWUE link between the universities of Karlsruhe and Stuttgart) are totally inadequate.

The telecommunication bandwidth required by a supercomputer is mainly proportional to its CPU speed. Table 3 shows the required speeds for transfer of different items.

application	required speed	present technique
text X-Windows	9.6 kBit/sec 20 kBit/sec	ISDN (64 kBit/sec)
color graphics file transfer (FTP) NFS	1-2 MBit/sec > 1 MBit/sec > 2 MBit/sec	Ethernet (10 MBit/sec)
simulation visualization (8 pictures/sec)	64 MBit/sec	FDDI (100 MBit/sec) ISDN-B (140 MBit/sec)
animation	1-10 GBit/sec	Ultranet (800 MBit/sec), frame buffer

Table 3: Speeds for transfer of different items

The problem of mass storage is even more limiting (University supercomputer disc storage capacity ~ 40 GByte, compare with expected output for advanced simulations) and there are few promising technological developments on the horizon.

The only hope to overcome these problems is to find out a new way in which we make use of supercomputers so as to effect a drastic reduction in the amount of data that needs to be stored or transferred. This requires a fundamental new kind of scientific simulation technique. The emergence of graphics super-workstation offers an opportunity to enable that essential change in methodology.
Today the typical solution sequence of a simulation includes the following steps:

Physical System description
Mathematical model
Vector Algorithm
Calculation
Analyze Results
Presentation of Results.

The handling of these steps in a sequential top-down manner must be replaced by a new simulation-technique using the capabilities of the graphic super-workstations. Graphics super-workstations can be utilized further "upstream" in the process as part of the calculation itself. Their specialized hardware and architectural properties can be used effectively in conjunction with the supercomputer in a distributed processing system

However, the availability of X-windows on workstations and supercomputers not only enables a user to watch the progress of a computation - and if necessary abort a bad run - but permits the user to interact the computation in process. The scientist can modify parameters, such as step size, grid spacing, damping terms, etc and also change the solution algorithm. Human interaction with the supercomputer by means of powerful graphics super-workstation will also enable or facilitate the solution of computationally difficult problems where the intervention of a human is a key or possibly essential part of the simulation. Examples include algorithms with different data storage techniques, systems containing multiple extrema where global extremum is desired and iterative processes that converge very slowly.

In these examples, the scientist becomes an essential part of the simulation. The user acts at a very high level by serving as part of a complicated nonlinear feedback loop, by using the knowledge of physical behavior not included in the program code or by detecting trends.

4. Technical aspects of "Distributed Applications"

The technical realization of distributed applications must be based on the existent hardware systems, the not yet fully standardized networks, the different operating systems and the different graphical standards. Standardization is a difficult task involving quite a number of organizations (ISO, IEEE, CCITT, ANSI, DIN), competing manufacturers and specialists and will take time and an still evolving market The requirements of the parts of the distributed system are shown in table 4.

supercomputer	communication services	workstation
operating system: UNIX System V Berkeley Extens.	remote login remote job entry remote monitoring remote printing	operating system: UNIX System V, Berkeley Extens., OSF/1
networks: Ultranet High speed I/O (HIPPI) FDDI Ethernet Channels (Hyper, IBM)	file sharing (NFS) file transfer (ftp) connecting (TCP, UDP /IP) auxiliary services	networks: Ultranet FDDI Ethernet
languages: FORTRAN + vector extensions graphical standards. X11, Motif GKS, PHIGS	distributed processing remote procedure calls (RPC) remote windowing network computing (NCS)	languages: FORTRAN, C, C++ (object orientated) graphical standards. X11, Motif, GKS, PHIGS, PEX, Starbase, HPGL different picture interchange formats: IGES, GIF, PCX, CGM

Table 4: Technical requirements

5. Conclusion

Today's supercomputers produce torrents of data, but the human brain still interprets numerical data as poorly as it always has. Scientists and engineers need an alternative to numbers and that alternative is images.

The overview of the characteristics and capabilities of currently available graphics super-workstations showed that one of the most important activities in a supercomputing environment was the analysis and interpretation of large masses of complex information.

The role of the graphics super-workstation in a supercomputing environment is absolutely essential to insure the integrity of analysis, to provoke insights and to empower the human as an essential component of the simulation.

ASPECTS OF BENCHMARKING FOR SUPERCOMPUTERS

Aad J. van der Steen
Academic Computing Centre Utrecht
Budapestlaan 6
3584 CD, Utrecht
The Netherlands

Abstract We address the problem of benchmarking supercomputers with the aid of synthetic programs. While not modelling an actual workload this approach has the advantage of providing more general information about the capabilities of the systems under consideration. We will review some benchmarking models and present the EuroBen benchmark as an example of a benchmark set that yields a performance profile which enables the identification of the strong and weak points of a machine in terms of a range of applications and their constituent algorithms.

I. INTRODUCTION

In this contribution we focus on the performance of supercomputers. A problem encountered with these systems is the large variety in their architecture. These can range from large vector computers with a limited amount of processors that share a common memory to machines with thousands of very simple processors and distributed memories. This can lead to an enormous *performance range*, sometimes potentially a factor of thousand or more, depending on the suitability of a certain piece of code for the underlying architecture.

There are several ways to deal with this problem, leading to different ways of benchmarking. Three main ways can be identified: *Theoretical benchmarking*, *Synthetic benchmarking*, and *Practical benchmarking* [12]. We will briefly discuss the various approaches and their particular advantages and drawbacks in the next three sections. In addition, in section V we will discuss one synthetic benchmark, the EuroBen benchmark, in more detail.

II. THEORECTICAL BENCHMARKING

Theoretical benchmarking is aimed at the modelling of the performance behaviour of machines with regard to their constituent components. These machine may or may not exist. In fact, this kind of modelling is often used to estimate trade-offs between performance and the application of more or different hardware components, more or less expensive technology, etc. Mostly, probabilistic models are used to estimate the influence of the various components. As supercomputers tend to be more complex than their more conventional counterparts the composition of theoretical benchmarks also is much more difficult and one often has to rely on simulators to acquire the desired information. This branch of benchmarking is more or less becoming a discipline in itself with journals like *Performance Evaluation* as an example of the interest in this field and a growing number of publications in other journals (see for instance [1], [5], [8], [11]).

An obvious advantage of the theoretical approach is one often is able to simulate machine- or component behaviour at a fraction of the time and price of actually building such a system. For existing machines a correct theoretical benchmark may adequately predict upperbounds and even performance profiles for various patterns of utilisation of the components.

A drawback is the difficulty of designing good theoretical benchmarks, especially for very complex behaviour large multi-user supercomputers. In most cases one has to be satisfied with the limited understanding obtained for isolated subsystems without being able to relate this to the total performance of the system. In addition, relatively small changes in the architecture of a machine may be difficult to incorporate in the machine model to be evaluated In all, theoretical benchmarking is a very difficult (but sometimes rewarding) approach to performance evaluation.

III. SYNTHETIC BENCHMARKING

Synthetic benchmarking consists of the running of one or more program kernels from which the performance of a system should be derived. The practice of running such synthetic programs is very easy in comparison to theoretical benchmarking or the practical benchmarking to be discussed later. This, and the fact that many "ready-made" benchmarks exist ([3, 10]), has made it very popular. Yet, although easy to run, the interpretation and the significance of the results for a particular test site are often far from straightforward [4, 6]. Especially with supercomputers one has to be very careful in addressing all aspects of the machine(s) at hand and one should refrain from judging such systems from just one parameter which supposedly would represent *the* performance of the system. Regrettably, this is a common practice with many vendors which in this way are able to "prove" the relative superiority of their systems. We will discuss better synthetic procedures in section V.

The advantages of synthetic benchmarking are already made clear. the testing procedure is relatively easy (although the design of a good synthetic benchmark is not). Synthetic benchmarking also is flexible. when new architectures emerge one can modify or extend an existing benchmark set easily to address new architectural features. In addition, when performed with care, the benchmark may yield more general information than just performance figures for particular programs.

Apart from the danger of misinterpretation that already has been mentioned, synthetic benchmarks cannot claim to give definive answers in selection procedures for a particular site. In this case one has to do additional benchmarking that more precisely reflects the workload of that site and possibly the variations in and evolution of this workload. So, although synthetic benchmarking may help in the preselection stage and in acquiring a general understanding, one has to complement it with site-specific testing procedures.

IV. PRACTICAL BENCHMARKING

Practical benchmarking is employed to obtain the specific answers concerning a systems appropriateness for a particular site. In this case a typical workload for that site can be run on such a system and usually one already has a detailed knowledge about configuration requirements and the total software environment that has to present. Because of the very specific nature of these benchmarks it is very hard to make general statements about the methodology beyond the most trivial ones: one should work with a representative workload, memory- and I/O-requirements should be properly heeded, etc. Consequently, there is almost no literature of general interest on practical benchmarking (a rare example is [9]). The large architectural differences between supercomputers will often frustrate practical benchmarking because it is impossible to port a complete workload to these systems in a way that allows simple comparison. This leads to the paradoxical situation that where a clear comparison is most needed it is often the most difficult to obtain. In this respect synthetic benchmarking may be of use to produce at least some partial answers.

V. THE EuroBen BENCHMARK

In June 1990 the EuroBen Group was founded [7] with the objective to distribute an easily portable synthetic benchmark set in Fortran 77 that would provide the user with a *performance profile* of the machine to be tested. The EuroBen benchmark is a European effort to bring some standardisation in the rather hectic field of synthetic benchmarking. As such it is certainly not the only effort. In this area both the PERFECT club [2] and the SPEC group should be mentioned. The latter communcates its results through its own newsletter.

The EuroBen benchmark is somewhat different from most benchmark sets in that the information on a machines' capabilities are obtained in a *hierarchical* way, i.e., the benchmark consists of modules of increasing complexity. The first module contains tests of basic operations, combinations of operations, and intrinsic functions while also issues like memory bank conflicts and memory contention are addressed. The second module contains basic numerical algorithms that mainly employ operations as tested in module 1. This enables to explain the performance of these algorithms in terms of the results from module 1. Module 3 contains more extensive algorithms that often combine several algorithms from module 2 (ODE- and PDE solvers, linear- and non-linear least square algorithms, etc.). Again the results from module 2 and module 1 can be used here for explanation of the results from module 3. Module 4 contains full application programs that again rely on the previous modules. Since early 1991 there is an effort from EuroBen and the PERFECT club to integrate their benchmarks in the sense that the PERFECT benchmark suite will act as module 4 in the EuroBen benchmark. This is appropriate as the PERFECT suite only contains application programs and aims at the same diversity in application areas as defined in the EuroBen model.

REFERENCES

[1] D.P. Agrawal, V.K. Janakiram, *Evaluating the Performance of Multicomputer Configurations*, IEEE Computer, May 1986, 23–27.

[2] M. Berry, et. al.,*The PERFECT Club Benchmarks. Effective Performance Evaluation of Supercomputers*, Int. Journal of Supercomputer Applications, Vol. 3, No. 1, 1989, 5–40.

[3] J.J. Dongarra, *Performance of Various Computers Using Standard Linear Equations Software in a Fortran Environment*, Argonne National Laboratory, Technical Report MCS-TM-23, Febr. 1991.

[4] J.J. Dongarra, J.L. Martin, J. Worlton, *Computer benchmarking, paths and pitfalls*, IEEE Spectrum, July 1987, 38–43.

[5] Ph. Ein-Dor, J. Feldmesser, *Attirbutes of Performance of Central Processing Units: A relative Performance Prediction Model*, Comm. of the ACM, Vol. 30, No. 4, 1987, 308-317.

[6] Ph.J. Fleming, J.J. Wallace, *How not to lie with statistics: the correct way to summarize benchmark results*, Comm. of the ACM, Vol. 29, No. 3, 1986, 218–221.

[7] A. Friedli, W. Gentzsch, R.W. Hockney, A.J. van der Steen, *A European supercomputer benchmark effort*, Supercomputer Vol. 6, No. 6, 1990, 14-17.

[8] M.A. Holliday, M.K. Vernon, *Exact Performance Estimates for Multiprocessor Memory and Bus Interference*, IEEE Trans. on Comp., Vol. C-36, No. 1, 1987, 76–85.

[9] K.E. Jordan, *Performance Comparison of Large-Scale Scientific Computers*, IEEE Computer, Mar. 1987, 10–23.

[10] F.H. McMahon, *The Livermore Fortran Kernels. a Computer Test of Numerical Performance Range*, Lawrence Livermore National Laboratory, Technical Report UCRL-53745, Oct. 1986.

[11] M.H. Schrage, *architectural barriers to array processor efficiency and their analysis*, Proc. of VAPP I, in Comp. Phys. Comm., Vol. 26, No. 3&4, 1982, 353–355.

[12] A.J. van der Steen, *Is it really possible to benchmark a supercomputer?*, in: A.J. van der Steen, ed., *Evaluating Supercomputers* (Chapman& Hall, London, 1990).

SUPERCOMPUTERS: THE SOFTWARE, VECTORIZING AND PARALLELIZING COMPILERS

Karim Roger K[illegible]er
Rechenzentrum der RWTH Aachen
Seffenter Weg 23
D-5100 Aachen
Germany

Abstract - Recently, the biggest increase in performance - measured in "naked" IPS or FLOPS - had been achieved by hardware developments and new machine architectures. That is why today the main attraction often is the architecture of new systems. In case of multiprocessing systems new results will be expected soon. Another aspect which becomes more and more important is the quality of software. For economical reasons standard software, e.g. the FORTRAN programming language, is prefered rather than special architecture-dependent or assembler languages. The benefits are easier implementation, better portability and the possibility to use universal programs and libraries on different machines. Portability for example is very important since the life cycle of application software lasts longer than the innovation time of hardware. Consequently the FORTRAN programming environments for some supercomputers from CRAY, IBM and Fujitsu are presented. The optimizing strategies of the compilers for vectorizing and parallelizing will be discussed but no valuation of the efficiency of the generated code will be made because this strongly depends on the underlying hardware. Besides, some language extensions, compiler-directives and analysing tools are presented. Operating systems will be focussed briefly since they build the missing link between runnable programs and hardware.

1. INTRODUCTION

After a general survey about the operating systems for supercomputers from CRAY, Fujitsu and IBM the compiler systems CRAY CFT 77, Fujitsu FORTRAN/VP and IBM FORTRAN/VS are presented and compared in detail. The machines from CRAY and the IBM 3090-x00 series are multiprocessor computers tightly coupled via shared memory. The older Fujitsu VP machines are monoprocessor machines but the new VP machines have a multiprocessor option and enhanced vector units. All machines are vector computers with some possibilities for parallelism provided by the shared memory. Although CFT 77 and IBM VS have some multitasking capabilities, their main goal is to vectorize rather than parallelize programs. The Fujitsu VP compiler has no multitasking features but a new one is announced which will support the new multiprocessor architecture.

In addition to these common FORTRAN systems that are extensively used in production environments two recent developments with better support for parallelism will be presented. CRAY's autotasking system CF77 and IBM's PARALLEL FORTRAN. Both systems have capabilities to parallelize work automatically and powerful extensions for explicit parallelism.

Vectorizing and parallelizing may be reguarded as program transformations which change the order of statements but don't change the program's semantics. If two statements share some input or output variables - called a data dependency - they must not be interchanged. Therefore the autovectorizing or autoparallelizing compilers analyse these dependencies before they change the order of statements. There are similar analysing and transformation methods for parallelization and vectorization.

2. OPERATING SYSTEMS

On IBM 3090 mainframes there are two major multiuser operating systems: MVS and VM. MVS is a multi-purpose operating system which has a wide functionality and is used in scientific and commercial environments as well, whereas VM is commonly used in scientific environments.

MVS (Multiple Virtual Storage) is designed to let each user have his own addressing room which is extremely protected against disturbance from outside of the user's world. Each user may start several tasks that are executed concurrently or in parallel System resources are managed dynamically by MVS. Thus, MVS supports parallelism in an efficient way.

VM virtualizes system resources. Each user [illegible] his own personal virtual machine (VM) which is separated from othe[illegible]s but, anyhow, the user may initiate communication with other VM[illegible]. These VM's send their requests to a control program (CP) which in turn carries out the requested operations Some VM's normally are dedicated as batch worker-machines to realize batch processing Parallelism is supported by assigning several VM's to a user application that are working independently.

VSP/I (Vector System Product/Interactive) for Fujitsu's vector machines is largely compatible with MVS/XA (user and operator-interface, job-control but not binary code) VSP/I has some extensions concerning vector processing on these special machines. For instance, programs under MVS usually operate in a virtual addressing mode but in VSP/I all vector programs run in a real mode the whole program code is resident in main storage Hence the vector processor never waits on program parts to be loaded from secondary storage.

The newest operating system for CRAY supercomputers is UNICOS which is based on AT&T UNIX SYSTEM V. UNICOS has some enhancements to UNIX for mainframe supercomputers, for example batch services, accounting tools and performance analysis tools. Above all UNICOS supports the multiprocessor architecture of CRAY machines.

3. STANDARD FORTRAN PROGRAMMING ENVIRONMENTS

3 1 THE COMPILERS CRAY CFT 77, FUJITSU VP, AND IBM VS

Now we take a closer look at the compilers CRAY CFT 77, Fujitsu VP, and IBM VS All these compilers are autovectorizing compilers based on FORTRAN 77 standard There are language extensions with CFT 77 modelled in FORTRAN 8X standard Information that cannot be extracted by syntactical analysis, for example vector lengths and true ratios of logical IF statements may be inserted as compiler directives into the source text. These directives will be treated like comments by other compilers. The compiler listing can be helpful when optimizing a program. All compilers show informations about the optimizations of loops, furthermore the IBM VS compiler presents all loops in a source to source modification. Hence this listing could be a vectorizing tool for other systems. Tools for run time analysis are available with all systems except IBM VS under VM/XA. The most comfortable is VECTUNE from Fujitsu which allows interactive analysing on loop and statement level. Optionally the results of this analysis may be taken to generate compiler directives automatically.

	CFT 77	Fuj. VP	IBM VS
auto-vectorizing	X	X	X
auto-parallelizing	cf 77	planned	-
programmed multitasking	X	planned	X
vector language extensions	8X	-	-
directives	X	X	X
analysing tools	X	X	only MVS

8X = FORTRAN 8X standard
cf77 = possible if cft 77 is embedded in cf77
Table 3.1.: compiler equipment

3.2. OPTIMIZING METHODS OF AUTOVECTORIZING COMPILERS

The quality of vectorizing and parallelizing compilers depends decisively on the ability of analysing and modifying loops. Nevertheless, loops with certain data dependencies after some optimization can be vectorized. For instance the loops with the data dependencies in Table 3.2.1. can be vectorized if the statements are exchanged.

```
      DO 10 I=1,N            DO 10 I=1,N
         ... = A(I)             A(I) = ...
      A(I+1) = ...                ... = A(I+1)
   10 CONTINUE             10 CONTINUE
```

Table 3.2.1.: data dependence, anti-dependence

For IF statements with an anti-dependence that cannot be reordered another vectorizing technique may be chosen. A temporary scalar is introduced which is expanded to a vector by the compiler (see Table 3.2.2)

```
   DO 10 I=1,N
     S    = A(I+1)
     A(I) = ...
     ...  = S
10 CONTINUE
```

Table 3.2.2.: anti-dependence after insertion of a temporary scalar

If a loop contains vectorizable and non-vectorizable statements vectorizable parts have to be separated from non-vectorizable to vectorize a portion of the loop. This is called partial vectorizing. Sometimes it depends on the value of a variable whether a loop is vectorizable or not. With the conditional vector method it is possible to generate sequential and vector-code and to decide at run-time which code is executed. In nested loops statements of the outer loop may be independent from those of inner loops. Then outer and inner loops may be splitted into two loops which are both candidates for vectorization (loop distribution). The vectorization of an outer loop is attractive if the inner loop is not vectorizable or the vectorization of the outer loop is simply more efficient, for example because of a greater vector length. This implies that the compiler is able to analyse whether loops can be interchanged (loop innermosting)

For a logical IF statement in a loop there are different methods for vectorization. Mask technique. a vector-bit-mask for the conditions will be built and only those right sides of assignments where the associated mask bit is true will be assigned to the left side. The disadvantage of this method is that all operands are sent through the pipeline, this would be waistful if the IF condition seldom is true In this situation the gather/scatter technique works more efficiently For all true conditions an index vector will be built and a special gather/scatter hardware accesses the operands indirectly with this vector and evaluates the right sides of equations. With the comp.ess/e pand method a vector-bit-mask is built as well but all operands are f .ched from memory avoiding indirect addressing. Then the fetched vectors are compressed and processed by constant stride hardware After processing the result vector will be expanded and stored. This method will work well if the ratio of load-store and arithmetical operations is small.

These described optimizing techniques for the compilers are listed in Table 3.2.3.

	CFT 77	Fuj. VP	IBM VS
statement reordering	cf 77	X	X
partial vectorizing	X	X	X
temporary scalars	X	X	-
conditional vectors	X	-	-
vectorizing outer loops	X	X	X
loop distribution	-	X	X
logical IF	M	M,G,C	M
Block IF	G	M,G,C	M
ELSE IF	G	M,G,C	M

M	=	mask operation
G	=	gather/scatter operation
C	=	compress/expand operation
cf77	=	possible if cft 77 is embedded in cf77

Table 3.2.3.: optimizing techniques

4. PARALLELIZING COMPILERS

4.1 GENERAL PARALLELISM CONCEPTS

Parallel execution of tasks for one application may improve its execution time but it increases the entire amount of work to be done (in CPU cycles) because there is a remarkable overhead generating parallel tasks, their synchronization and communication. Therefore, parallelism must be handled carefully. However, if one application covers most of the main storage it should be processed in parallel since it would be waistful to execute it on only one CPU because of other CPU's running idle. In general to maximize throughput the share of main storage and processing power should be equal. Another motivation for parallel processing may be that an application may be done only in realistic time by parallel speed up or the application has a certain deadline, e.g. meteorological calculations.

There are two classes of multitasking concepts:

- Fine-grain parallelism means parallel execution of small fractions of code, for example different chunks of a DO loop
- Coarse-grain parallelism signifies parallel execution of progr.m segments with larger granularity, for example subroutines

Up to now techniques for interprocedural data dependency analysis to allow automatic coarse grain parallelization are in the experimental phase. The programmer has to introduce explicit parallelism by using a multitasking interface, for example library calls, or new language elements or by specifying compiler directives. All interprocedural data dependencies have to be synchronized explicitly by the programmer. Within one subroutine compilers may find small independent parts of code (fine-grain parallelism) Thus, no restructuring of the algorithms will be needed. In coarse-grain parallelism it is a big problem to partition work into portions of approximately equal size so that the work load is balanced among the processors. Fine-grain parallelism helps load balancing by filling the gaps of processor idle time owing to the existence of many small computational units. If the sum of idle times of processors is greater than that for initiation,communication and synchonization of parallel work the application is accelerated as well as system throughput is increased. Vectorization and parallelization may exclude each other, for example a long vectorizable DO loop may only be splitted into smaller chunks of parallel loops.

Here are some advices for parallel programming:

- at first optimize the program for one processor (vectorization)
- choose equal granularity of tasks (load balancing)
- coarse-grain parallelism should only be used for tasks with more than 10 000 operations because of the overhead involved with it
- use few multitasking routines and parall lize the hot spots of execution time

4.2. CRAY AUTOTASKING SYSTEM cf77

CRAY provides three concepts for multitasking: macrotasking, microtasking and autotasking As macrotasking is the coarse-grain mechanism microtasking and autotasking are fine-grain. With autotasking automatic fine-grain parallelization in subroutines is introduced. Autotasking, microtasking and macrotasing can coexist in different subroutines but not in a single one.

MACROTASKING

Macrotasking is available by means of FORTRAN subprogram calls from a special multitasking library and offers high level synchronization and communication primitives. These subroutines are the interface to the library scheduler which manages the single tasks, the synchronization between them and performs requests to the operating system. The library scheduler does not use hardware for most synchronization primitives. Thus for small granularity the activities of the library scheduler can lead to an extensive overhead: Macrotasking should be used with large granularity.

THE MACROTASKING LIBRARY consists of.

- routines to manipulate tasks
- routines to control events for synchronization
- routines to control critical regions which can be executed by one processor exclusively at one time

MICROTASKING

Microtasking offers much faster basic functions for synchronization and communication by accessing special hardware and is realized by compiler directives (CMIC$) The CMIC$-directives are translated into multitasking library calls.

THE MICROTASKING DIRECTIVES consist of:

- directives requesting microtasking mode and logical CPU's
- directives to define microtasking control structures (MCS)
- directives to mark critical regions

With the GETCPUS directive logical CPU's are created Logical CPU's are in a busy loop waiting for work If there is some work to do all idle CPU's are able to execute parts of it The operating-system scheduler takes this busy loop concept into account· logical CPU's which terminate working are rescheduled and are immediately ready to work again. Therefore microtasking is very efficient Within one MCS several parallel processes may be defined each of them executed exclusively by one logical CPU. Processes within one MCS are scheduled dynamically in unpredictable order and therefore must be independent. Data dependencies may be considered by putting the corresponding processes into different MCS's, since a MCS must be terminated before the program continues. Code outside MCS's can be processed in parallel by all logical CPU's in unpredictable order.

AUTOTASKING

Autotasking is realized by using some microtasking- and new autotasking-CMIC$-directives.

The CRAY autotasking system mostly consists of well-known components for instance the cft77 compiler. Other new parts are the preprocessor fpp and the multitasking translator fmp.

fpp analyses the data dependencies and transforms the source code to modified source code so it can be vectorized or parallelized or is simply more optimal in scalar. The modified source code contains directives for vectorizing and multitasking. Automatic parallelizing may be switched off so fpp may create pure vector source code. Depending on the CRAY architecture vectorization is prefered rather than parallelization. fpp only analyses DO loops, which are the most frequent type for parallel or vector constructs. All loops are vectorized if possible. In nested loops the outermost loop will be parallelized if possible and the innermost loop will be vectorized. By an option vectorizable loops with high iteration counts may be splitted into several parallel loops and each parallel loop will be vectorized. Even the outer loops of non-vectorizable inner loops may be parallelized. fpp provides inline-expanding of subroutines so a more global analysis of data dependencies can be made. But it should not be overdone because then the code grows immense. If no additional loops became vectorizable which should be the case in good vector programs the gain of run-time is little.

If no parallelism is requested the modified source code program may directly be compiled with cft77. However to exploit parallelism the output of fpp must be processed by fmp before being translated with cft77. fmp changes autotasking directives to machine dependent library calls.

4.3 IBM PARALLEL FORTRAN

Parallel FORTRAN has three fundamental extensions for fine- and coarse-grain parallelism on IBM 3090 multiprocessors.

- extensions to the compiler for automatically generating parallel code
- language extensions for explicit parallelism
- extensions to the library for synchronizing parallel execution

Fine-grain parallelism is supported by language-extensions and automatical detection of implicit parallelism, coarse-grain by language and library extensions.

The Parallel FORTRAN environment consists of the parallel application program, some FORTRAN processors and the parallel library. The FORTRAN library maps the parallel pieces of work onto virtual processors called FORTRAN processors. The operating system maps FORTRAN processors onto real machine processors. The implementation of FORTRAN processors depends on the operating system. FORTRAN processors under MVS/XA are tasks. Under VM/XA several virtual machines build a virtual multiprocessing machine and each FORTRAN processor is executed by one virtual CPU. The maximum degree of parallel execution can be varied at run-time by specifying an option for the number of virtual FORTRAN-processors. The user can define more parallel pieces of work than the number of FORTRAN processors allows. Additional pieces of work build a queue and finishing FORTRAN processors take their work from this queue. As long as the queues are filled the FORTRAN processors can continue working without the operating system. Only if a queue empties or refills there is overhead of the operating system by suspending or restarting a FORTRAN processor. There is no operating system overhead to schedule and synchronize parallel work.

AUTOMATIC PARALLELISM

The compiler determines whether it is cost effective to execute the loop in parallel, otherwise vector or scalar code is generated. A loop selected for parallel execution may contain inner loops that are parallelized, vectorized or serial. A loop selected for vector operation may contain only inner loops which are serial. There are for example directives to indicate a preference for parallel, vector or serial code and the number of iterations to be grouped together as a unit of work.

LANGUAGE EXTENSIONS

Parallel FORTRAN provides two types of language extensions which both can be nested and intermixed:

- in-line extensions: define parallelism within a routine
- out-line extensions: define parallelism across routines

In general in-line extensions support fine-grain parallelism as out-line extensions provide coarse grain parallelism.

IN-LINE EXTENSIONS

There are two new in-line extension language elements the PARALLEL LOOP and the PARALLEL CASES. Each iteration of a PARALLEL LOOP can be executed concurrently and the order of execution of the iterations is not guaranteed. Parallel executed blocks of statements are declared by PARALLEL CASES. Such blocks may contain straight code, parallel or vector loops. Although, PARALLEL CASES is an in line construct this primitive should better be used for coarse-grain parallelism because in current implementations there is a large overhead involved with it. Thus . It is possible to number cases to define an execution order among blocks of statements. Every acyclic graph of data dependencies may be transformed into PARALLEL CASES.

OUT-LINE EXTENSIONS

Out-line extensions create disjoint, asynchronous execution environments called tasks. A task is a collection of subroutines that is independent of other tasks. This task concept is hierarchical. In the beginning there is one root task which may create several child tasks with an ORIGINATE statement. A task is terminated with a TERMINATE primitive or automatically at its end. SCHEDULE or DISPATCH assign the execution of a subroutine to a task. As dispatched tasks run completely asynchronous scheduled tasks require a WAIT FOR instruction to terminate. Other primitives for subtask synchronization are locks and events, which are recommended because explicit WAIT FOR is very expensive. Locks ensure that only one processor at a time gets access to a certain resource. Events may be used to make a task wait until another task has reached a certain point of execution. With locks and events self balancing dynamical workloads can be created. PARALLEL LOOP and PARALLEL CASES constructs are automatically load balanced by the library. Only if the user explicitly shares data between tasks the execution of tasks can affect one another.

5. CONCLUSION

Today vectorizing is an established method. The three described compilers all are able to vectorize automatically but in spite of that they are very different. The CRAY CFT compiler is the successor of the older CFT 1.15 compiler. Since the aged CFT 1.15 was not based on theoretical program analysis, CFT has been designed completely new. The IBM FORTRAN/VS compiler is based on an older sequential version. The biggest handicap for VS is the underlying 3090 cache-hardware. If the accesses run out of cache-memory (for example row accesses in matrices) vectorizable loops are not vectorized by the compiler. The newest version of FORTRAN/VS Rel 2.5 is an autovectorizing and autoparallelizing compiler. Supplementary to enhanced vectorizing capabilities it includes features from PARALLEL FORTRAN. Fujitsu's VP compiler adapts the structure of vector registers to the program structure given. It has three vectorization methods for IF statements (mask, gather/scatter, compress/expand). The announced Fujitsu FORTRAN/VP-EX compiler will adapt programs to the new hardware and improve the performance of programs. For instance, the vector-pipeline is used inefficiently if the DO loop does not contain enough operations. Loop unrolling expands the executable statements in a DO loop and reduces the iteration count ,thus, pipelines can work more efficiently. Program performance is increased further by integrating subroutines into calling routines and optimizing the modified program in a more global manner. To exploit the new multiprocessor architecture the compiler FORTRAN/PP will be developed with automatic intraprocedural parallelism and language extensions for interprocedural parallelism.

There is no ideal compiler, all are in continous development and in addition to that often have to be adapted to new hardware.

With parallelization however new problems arise, for instance for coarse-grain parallelism synchronization has to be programmed explicitly. To justify the overhead derived from parallelization work has to be partitioned in large parts of code. This in fact conflicts with dynamic load-balancing. If the programmer is aware of his problem and of the underlying hardware too, many errors may be avoided . Hardware aspects are important since, for instance, the shared memory systems focussed here not only use memory to store data but also to communicate between tasks. Parallelizing compilers today are in the age of maturity. Possibly the next generation will be able to analyse data dependencies above the DO loop level Analysing results will be displayed and some sort of interactive semi-automatical parallelizing will be realized.

6. ACKNOWLEDGEMENT

I would like to thank W. Juling who made valuable contributions to this paper.

7. BIBLIOGRAPHY

Detert, U., "Programmiertechniken für die Vektorisierung," Praxis der Informationsverarbeitung und Kommunikation 3/87, October 1987

Dubrulle, A.A., Scarborough, R.G., Kolsky, H.G., "How to write good vectorizable FORTRAN," Palo Alto Scientific Center Report No. G320-3478, September 1985

Gentzsch, W., "Parallelverarbeitung auf den Rechnern von CRAY, IBM, ALLIANT und CONVEX," Praxis der Informationsverarbeitung und Kommunikation 3/89, October 1989

Hege, H.-C., 'Datenabhängigkeitsanalyse und Programmtransformationen auf CRAY-Rechnern mit dem Fortran-Präprozessor fpp," Technical Report TR 90-4 Konrad-Zuse-Zentrum für Informationstechnik Berlin, February 1990.

Nagel, W.E., Szelenyi, F., "Multitasking on Supercomputers: Concepts and Experiences," IBM European Center for Scientific and Engineering Computing, Rome, Italy, Report Number: ICE-VS05, May 1989

Perrot, R.H., Zarla-Allabadi, "Supercomputer Languages," Computing Surveys, Vol. 18, No. 1, March 1986

Toomey, L.J., Plachy, E.C., Scarborough, R.G., Sahulka, R.J., Shaw, J.F., and Shannon, A.W., "IBM Parallel FORTRAN," IBM Systems Journal 27 (4) 416-435, 1988

Dataflow Programming Languages

Rishiyur S. Nikhil
Laboratory for Computer Science
Massachusetts Institute of Technology

545 Technolology Square, Cambridge, MA 02139, USA

Introduction

Dataflow languages are attractive for parallel computation because of their expressive power— their high-level, machine-independent, implicitly parallel notation— and because of their fine-grain parallelism, which seems essential for effective, scalable utilization of parallel machines.

Dataflow languages are no longer tightly coupled with dataflow architectures. Today, we understand the compiling issues for dataflow languages well enough that we can recommend these languages to programmers without having to rely on architectural support.

There are several dataflow languages described in the literature, including Id [7], Sisal [6], and Val [1]; due to space limitations, we will base our discussions here on Id.

Expressive Power of Dataflow Languages

"Expressive power" does not have a precise definition; one can make reasoned arguments about why one language is more expressive than another, but ultimately the judgment is subjective. Theoretically, FORTRAN and assembly language have the same computational power, but almost everyone would agree that FORTRAN is more expressive, because it is architecture-independent and because it requires less detailed specification of problem solutions and is "closer" to the way human programmers think about problems.

Dataflow languages are based on *functional programming languages.* Unlike most programming languages that have evolved upwards from machine languages, functional languages are based on the λ-calculus, a mathematical theory of functions that predates digital computers [3].

Functional notation:

An aspect of the expressive power of functional languages is that one can manipulate complex objects (entire data structures, and even functions themselves) as ordinary values. Consider this function:

```
def map2 f A B = ... details omitted ...
```

taking 3 arguments: a function `f` and two arrays `A` and `B`, and returning as its result a new array (call it `C`) where

```
C[j] = f A[j] B[j]
```

i.e., it applies `f` to each pair of components of `A` and `B` and returns an array containing the results. (In Id and other functional languages, it is common to use a "curried" notation, omitting parentheses, commas, semicolons *etc.* around the arguments of a function.)

Now, we can specialize `map2` to define a vector sum function:

```
def vsum A B = map2 (+) A B ;
```

Here, the function "+" is supplied as the `f` parameter to `map2`, so that `vsum` adds the vectors `A` and `B` pointwise, and returns a vector containing the sums. Further, since `vsum` is itself an addition function, we can perform vector addition on vectors whose components are themselves vectors:

```
def vvsum AA BB = map2 vsum AA BB ;
```

Another useful higher order function is:

```
def foldl f z A = ... details omitted ...
```

which computes the "`f`-reduction" of array `A`, where `z` is the zero of the function `f`, *i.e.,* it computes:

```
f ...(f (f z A[1]) A[2]) A[n]
```

Using `foldl`, one can express the inner-product of two vectors:

```
def ip A B = foldl (+) 0 (map2 (*) A B) ;
```

Here, `map2` multiplies `A` and `B` pointwise, and `foldl` computes the sum of the resulting vector of products.

The ability to manipulate entire complex objects and functions themselves as values relieves the programmer of the tedium of descending down to explicit loops and assignments at every stage— such details can be hidden in a few generally useful functions such as `map2` and `foldl` [2]. Such higher order functions are the "power tools" of functional programming, in the sense that they amplify the programmer's effort.

Implicit parallelism:

Another aspect of expressive power is implicit parallelism. In most parallel languages, the user must carefully orchestrate parallelism: explicitly partition the program into parallel processes, specify the placement of data, specify the placement and scheduling of processes, *etc.* Often, these details are architecture specific, and sometimes even configuration specific, resulting in fragile, non-portable code.

In functional languages (and in all our examples above) we do not have to specify what can be done in parallel. In principle, everything can be performed in parallel, limited only by data dependencies. For example, the implementation of `map2` may perform all the applications of `f` in parallel. Thus, functional languages do not increase the complexity of the programmer's task when moving from a sequential to a parallel environment.

M-structures: implicitly synchronized state:

In pure functional languages, there is no concept of "state," *i.e.,* structures whose contents change during program execution. There are no assignment statements— programs are functions that compute new values from old. While many algorithms can be expressed clearly in this style, there are some problems.

One difficult area is. algorithms making effective use of non-determinism. Functional languages have the Church-Rosser property [3] which guarantees that the answer produced by a functional program does not depend on the particular parallel execution structure chosen by an implementation. This is usually a major advantage; unlike other parallel languages, functional programs are not subject to *race conditions* leading to non deterministic and irreproducible behavior (which complicates debugging).

However, consider the problem of traversing a directed graph to count the number of nodes reachable from a given root node. The following strategy seems simple: starting at the root node, fan out

Funding for this project is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-89-J-1988.

in parallel on all outgoing edges from each visited node. To avoid repeated traversals of shared subgraphs (and cycles), the first traversal to reach a node marks it "visited," so that subsequent traversals reaching the node via other paths will observe the mark and retreat. Non-determinism is exploited because, of all the incoming traversals arriving at a shared node, we do not care which one arrives first to mark the node and traverse its successors. There is no way to express this idea in a functional language one must pick a deterministic order to traverse the nodes. This not only clutters up the program significantly, but also sequentializes it.

Introducing non-determinism into a functional language is not very different from introducing state— constructs for one can be used to simulate the other. Not surprisingly, functional languages also have difficulty in dealing with input output, which seems quintessentially linked to the notion of state.

In Id, the functional core is extended with *M-structures* which are updatable data structures. Unlike other languages where updates are protected by separate synchronization primitives (*e.g.*, semaphores), accesses to M-structures in Id are combined with implicit synchronization. For example, a component of an M structure array (an "M_array") can be incremented using the statement:

```
A![j] = A![j] + 1
```

The expression `A![j]` on the right hand side not only reads, but simultaneously locks the location. The assignment "`A![j] = ...`" not only writes the value back, but simultaneously unlocks the location. This guaratees that the increment is *atomic*, *i.e.*, even if several computations execute this statement concurrently, the increments will be done properly. Coupling accesses with synchronization in this manner leads to clear and concise programs [4].

Implementations and performance

Dataflow implementations have become increasingly sophisticated in recent years.

Compilers for dataflow languages do not have the problem of detecting parallelism; rather, they often have the opposite problem of trying to contain excess parallelism. Without optimization, dataflow programs can be quite voracious in their resource demands. For example, a direct interpretation of the `map2` function would allocate a new vector for each result that it computes. In general, this storage must be allocated dynamically, and it is necessary to have a garbage collector to recycle memory occupied by structures no longer in use. A related problem is computational cost. If we have a "point" represented as a vector of 3 values (x, y, z), and we wish to displace it one unit along the x-axis, functional semantics demands that we produce a new vector containing ($x + 1$, y, z). Notice that we have to copy y and z. These overheads can be quite high. Thus, reuse of storage is essential for efficient implementations.

For Sisal, which is a pure functional language, researchers have developed program analysis techniques that allow the implementation to reuse storage heavily [9]. Sisal researchers report that on the Livermore loops, they are now able to compile code for Cray supercomputers that is competitive with FORTRAN codes.

The Id compiler [10] is unable to use the Sisal analysis techniques directly because of higher order functions, non-strictness and recursive data structures. However, it is a highly optimizing compiler that uses lifetime analysis, loop bounding and several other novel optimization techniques. The Id compiler is currently targeted to the Monsoon dataflow machine that is being built as a collaboration between MIT and Motorola, Inc. At time of this writing (March 1991), complete uniprocessors have just become operational, and 8-processor Monsoon machines are expected in April 1991. We have run only a few small benchmarks (such as matrix multiplication), but it appears that Id on Monsoon can be competitive with C on a modern workstation.

Fine grain parallelism seems essential if we are to simultaneously achieve the goals of general-purpose parallel programming and scalable performance. As processors become faster and machines become more parallel, memory latencies are becoming relatively larger. A general way to address this is rapid, fine-grain thread switching, *i.e.*, each processor needs a large pool of threads amongst which it can be efficiently multiplexed so that it can always do useful work while some threads are waiting (for long-latency memory requests, for synchronization, for results of procedures, *etc.*).

Dataflow languages and their compilers offer a systematic, complete approach towards this computational model. New compilers are under construction at MIT and Berkeley that use dataflow principles to compile Id even for non-dataflow machines [8, 5]; these compilers should enable Id to become widely available.

Conclusion

There is an interesting parallel between parallel programming today and sequential programming in the early 1950's. At that time, programs were written in assembly language, and programmers had to be acutely aware of the architectural details of their machines. There was widespread skepticism that high-level, machine-independent programming was possible with acceptable efficiency. John Backus and his group changed all that, with their outstanding FORTRAN implementation. Today, parallel programmers have to be acutely aware of the architectures of their parallel machines, and there seems to be widespread skepticism that high level, machine independent parallel programming is possible with acceptable efficiency. Our hope is that dataflow languages like Id can do for parallel programming what FORTRAN did for sequential programming. The prospects look bright.

References

1. W. B. Ackerman and J. B. Dennis. VAL– A Value-oriented Algorithmic Language: Preliminary Reference Manual. Techn. Rep. TR-218, MIT LCS, 545 Tech. Sq., Cambridge, MA 02139, June 1979.
2. J. Backus. Can Programming be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs. *CACM*, 21(8):613–641, August 1978.
3. H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics.* North Holland, Amsterdam, 1981.
4. P. Barth, R. S. Nikhil, and Arvind. M-Structures. Extending a Parallel, Non-strict, Functional Language with State. Tech. rep., MIT LCS, 545 Tech. Sq., Cambridge, MA 02139, March 1991. Submitted for publication.
5. D. E. Culler *et. al.* Fine-grain Parallelism with Minimal Hardware Support. A Compiler-Controlled Threaded Abstract Machine. In *4th Intl. Conf. on Arch. Support for Prog. Langs. and Op. Systems*, April 1991.
6. J. McGraw *et. al.* SISAL Reference Manual. Tech. rep., Lawrence Livermore Natl. Lab., 1984.
7. R. S. Nikhil. Id (Version 90.0) Reference Manual. Tech. Rep. CSG Memo 284-1, MIT LCS, 545 Tech. Sq., Cambridge, MA 02139, USA, July 1990.
8. R. S. Nikhil. The Parallel Programming Language Id and its Compilation for Parallel Machines. In *Proc. Wkshp on Massive Parallelism, Amalfi, Italy, October 1989.* Academic Press, 1990 (to appear).
9. J. E. Ranelletti. *Graph Transformation Algorithms for Array Memory Optimization in Applicative Languages.* PhD thesis, Lawrence Livermore Natl. Lab., 1987.
10. K. R. Traub. A Compiler for the MIT Tagged-Token Dataflow Architecture. Tech. Rep. LCS TR-370, MIT LCS, 545 Tech. Sq., Cambridge MA 02139, August 1986.

Monsoon: Dataflow Architectures Demystified

Kenneth R. Traub
Motorola Cambridge Research Center
One Kendall Square, Building 200
Cambridge MA 02139 USA

1 Introduction

Dataflow architectures often seem mysterious compared to more familiar von Neumann computers. This reflects the history of their evolution, which has tended to emphasize tokens flowing on dataflow graphs as the metaphor for their operation. Recent advances in dynamic dataflow architectures, as well as in our understanding of them [4], make it possible to see dataflow machines in a more conventional light. While the tokens-on-a-graph metaphor is still a useful way to reason about the machine, and especially about a particular compilation style for the machine, one can also think of a dataflow program as multiple, interacting sequential threads [6]. This makes a dataflow machine look more like a typical multiple-instruction, multiple-data (MIMD) multiprocessor, and highlights its strengths relative to other MIMD architectures.

This article describes the Monsoon architecture from this point of view. Monsoon [5, 2] is a prototype dataflow computer currently under construction in a joint effort between the Massachusetts Institute of Technology and Motorola, Inc. Most of the discussion directly applies to any dynamic dataflow computer with an explicit token store, including the Sandia Epsilon-2 [3], and the ETL EM-4 [7].

2 Parallel Machine Language

Dataflow architectures are perhaps the first general purpose computers that execute a *parallel machine language*. It is a parallel machine language because it has primitive operations for managing parallel computation, namely, *fork* and *join*. A *fork* instruction initiates a new, independent thread of computation in the machine, with its own program counter and register set. This is depicted in Figure 1a; as the figure suggests, *fork* is like both taking a branch and also continuing on to the next instruction. A *join* instruction brings two independent threads together into a single thread. Each thread will try to execute the *join* instruction (by falling through to it or by jumping to it), but only the second to arrive will continue with the next instruction following the *join*. This is depicted in Figure 1b What is unique about the dataflow architecture is that these primitives are single instructions, whereas in conventional multiprocessors they must be simulated through software procedures that can require hundreds or thousands of instructions for each *fork* or *join*.

What happened to dataflow graphs? They emerge from a particular compilation strategy that introduces many *fork* and *join* instructions, as illustrated in Figure 1c. This reveals the correspondence between multi-thread sequential code and dataflow graphs: a chain of one-input, one-output dataflow operators is like a sequence of conventional von Neumann instructions. The token flowing along such a chain simply contains the processor state for that thread: a program counter (in dataflow parlance, the instruction pointer part of the "tag"), and the thread's register set. In Monsoon, that set is really only a single general register, called the "value," and a special frame pointer register (another part of the "tag" in dataflow jargon) used as the base of an addressing mode for accessing data local to a procedure activation.[1] A *fork* is a one-input, two-output dataflow operator, and a *join* is a two-input, one-output operator.

The instruction set of Monsoon includes single instructions that do a *join*, an arithmetic operation, and a *fork*. Such an instruction would be drawn as a two-input, two-output arithmetic node in a dataflow graph. Having such instructions means Monsoon can efficiently execute "pure dataflow" graphs, but again, this is only one of many compilation strategies possible.

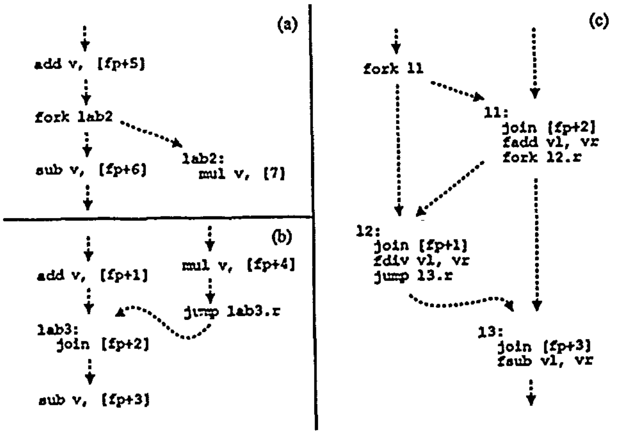


Figure 1: Examples of *fork* and *join*

3 Split Phase Memory Transactions

Essential to any multiprocessor is the ability to tolerate long memory latency [1]. Dataflow architectures do this through *split-phase* memory transactions [1]. From the program's point of view, a split-phase *fetch* instruction to fetch from location L behaves like a *fork*: the thread issuing the *fetch* continues processing, and a new thread is initiated. The new thread will have its value register initialized to the value fetched from L. In the machine, the *fetch* instruction actually operates by sending a special request message to the processor containing L (this is the first phase), and continuing on to the next instruction in the thread. The remote processor sends back a response message (this is the second phase), which initiates the new thread in the processor that originally executed the *fetch*. This new thread may begin a totally independent computation, or synchronize with the original thread through a *join* instruction. As long as there is a sufficient supply of parallel activity (*i.e.*, other threads previously *fork*ed), the processor will not be idle between the issuing of the request and the receipt of the response.

A split-phase *store* instruction is similar, except that the response from a *store* carries no value, only an acknowledgment that the store is complete.

4 Synchronization

Equally essential to any multiprocessor is the ability to rapidly synchronize parallel computations [1]. The *join* operation already discussed is the primary means of synchronization in dataflow architectures. In "explicit token store" dataflow architectures such as Monsoon, this operation is performed quite efficiently through the addition of a few *presence bits* to each word of an ordinary data memory. Every *join* instruction names a word in that memory, called the *rendezvous point*. Typically, the addressing mode used to name that rendezvous point is a fixed offset relative to the activation frame (see Section 7). Initially, the presence bits on the rendezvous point are *empty*. When the first of the two joining threads executes the *join*, the contents of that thread's value register are stored in the data part of the rendezvous point, and the presence bits are changed to *present*. The first thread then "dies," meaning that the processor goes on to execute some other thread, created by an earlier *fork* or response to a split phase transaction. Some time later, another thread executes a *join* which names the same rendezvous point, finding the presence bits set

[1] Monsoon actually provides more than one general register along chains of instructions, with the restrictions that the additional registers are uninitialized in new threads created via *fork*, and that the additional registers are destroyed when the thread executes a *join*.

to *present*, that thread continues with the next instruction following the *join*, and the synchronization is complete. In the case where the *join* is part of a two-input arithmetic instruction, the value registers of the two joining threads become operands, one of them having been recorded in the rendezvous point.[2]

Notice that the *join* operation requires no exotic hardware such as associative memories, only a few extra bits. Notice, too, that every word of data memory comes equipped with presence bits, so there is an effectively unlimited namespace for synchronization events, or, putting it another way, an unlimited number of synchronizations may be in progress at once.

5 Synchronizing Memory Operations

The *join* operation provides one means for computations to synchronize; another is provided by synchronizing memory operations. These exploit split-phase transactions and presence bits to provide a variety of ways to synchronize producers of data structures with their consumers. An example are the split-phase operations *I-fetch* and *I-store*, which synchronize many readers of a location with a single writer. Before an *I-store*, the presence bits of the location must be *empty*; a subsequent *I-store* request will set them to *present* as well as storing the data. An *I-fetch* request to a *present* location behaves like an ordinary *fetch*. An *I-fetch* request to an *empty* location, however, sets the presence bits to *deferred* and does not generate a response. A subsequent *I-store* request will store the value and set the presence bits to *present*, as before, and also will produce the response to the deferred *I-fetch*. (How the information to generate the responses is maintained is beyond the scope of this paper; see [2].) Thus, the consumer may issue an *I-fetch* without knowing whether the data has been stored: the response will not be generated until after the corresponding *I-store* has happened. This sort of usage generally involves two types of synchronization: the memory synchronization as described above, and a *join* back at the issuing processor to synchronize the response to the *I-fetch* with the thread that issued the request.

Monsoon provides several forms of synchronizing memory operations, including *take* and *put* operations for building locks; different operations are selected by choosing an appropriate *fetch* like or *store* like opcode as the first phase. Of course, ordinary imperative *fetch* and *store* operations are provided, their second phases of these operations simply ignore the presence bits entirely.

6 Compiler Controlled Thread Granularity

It is often said that parallelism in dataflow architectures is fine-grained. In fact, the compiler has considerable control over the granularity of parallelism exposed to the hardware. Thread granularity can be measured by the number of instructions a thread can execute before it stops or synchronizes with another thread via a *join*. Now consider compiling an expression like `(c*a[i]) + (c*b[i])`. Assuming `a` and `b` are global arrays, the fetches of `a[i]` and `b[i]` will be split-phase transactions. There are two ways the compiler could express the remainder of the computation: (1) it could *join* the two responses, then multiply each by `c` and add the products; (2) it could have a separate thread for each response, each of which multiplies, and then those two threads *join* together and add. The arithmetic in strategy (1) is in one large thread, whereas in (2) it is in three smaller threads.

The traditional dataflow style of compilation uses arithmetic instructions that always join, and so that style is a fine-grain style of compilation. This is a strategy well matched to non-strict functional languages such as Id, as their semantics limit the size of grains that any compiler could discover [8]. Larger grains might be employed by a compiler for an imperative language. What is interesting is that because *fork* and *join* are so efficient, the choice of large grains versus small grains is often determined by addressing modes and register allocation, as opposed to the overhead of synchronizing. As Monsoon's addressing modes are tuned to the dataflow style of compiling, finer grains can in many cases result in superior code, even though there is more joining.

7 Compiler Controlled Task Distribution

A common misconception is that work distribution across processors in a dataflow machine is at the level of individual instructions, *i.e.*, at every *fork*. In fact, the thread created by *fork* always executes on the same processor that executed the *fork* instruction itself. So strictly speaking, *fork* does not create any *parallel* activity, only another thread that is queued up until the processor finishes with the current thread.[3] Creating a thread on another processor requires a special kind of *fork* instruction called *send* or *start*, hence, the granularity work distribution is under compiler control. Typically, a compiler inserts *send* instructions as part of a procedure calling convention: a procedure is invoked by creating a new activation frame (stack frame) for the call—the frame may be on a remote processor—and then using *send* to start one new thread on that processor for each argument, where the value register of each new thread is a parameter for the call. Similarly, the called procedure returns a result to the caller by using *send* to start a thread back on the calling processor, which may then join with the thread that initiated the call in the first place. Task *distribution* is at the level of procedure calls, even though instruction *scheduling* is at the finer granularity of threads.

It has been tacitly assumed up to now that local variables of a procedure invocation, as well as the rendezvous points for its *joins*, are collected together in an activation frame associated with the invocation. (Activation frames are like stack frames in a uniprocessor, except that because of the possibility of parallel procedure calls they form a tree instead of a stack.) This is accomplished through the use of addressing modes that operate relative to a *frame pointer* that is part of the register set of every thread. One of the operands to the *send* instruction is a frame pointer for the new thread; for a procedure call, this would be a pointer to the new activation frame created for the call. The actual routing of the new thread to the target processor is accomplished simply by considering the most significant bits of the frame pointer to be a processor ID.

8 Summary

Though dataflow architectures have grown up in a tradition of radical departure from von Neumann computing, they have progressed to the point where they may be seen as an evolutionary step. The current direction in dataflow processor design is to exploit the new-found similarity to conventional architectures, leading to processors that match von Neumann efficiency on sequential code while retaining the benefits of a truly parallel machine language.

[1] Arvind and R. A. Iannucci. Two fundamental issues in multiprocessing. In *Parallel Computing in Science and Engineering*, volume 295 of *LNCS*, pages 61–88. Springer-Verlag, Jun 1987.

[2] D. E. Culler and G. M. Papadopoulos. The explicit token-store. *J. Par. and Dist. Comp.*, 10(4):289–308, Dec 1990.

[3] V. G. Grafe and J. E. Hoch. The epsilon-2 multiprocessor system. *J. Par. and Dist. Comp.*, 10(4):309–318, Dec 1990.

[4] R. S. Nikhil and Arvind. Can dataflow subsume von Neumann computing? In *Proc. 16th Ann. Int. Symp. on Comp. Arch.*, pages 262–272. IEEE, Jun 1989.

[5] G. M. Papadopoulos and D. E. Culler. Monsoon. an explicit token store architecture. In *Proc. 17th Ann. Int. Symp. on Comp. Arch.* IEEE, 1990. (To appear).

[6] G. M. Papadopoulos and K. R. Traub. Multithreading. A revisionist view of dataflow architectures. In *Proc. 18th Ann. Int. Symp. on Comp. Arch.* IEEE, May 1991. (To appear).

[7] S. Sakai, Y. Yamaguchi, K. Hiraki, Y. Kodama, and T. Yuba. An architecture of a dataflow single chip processor. In *Proc. 16th Ann. Int. Symp. on Comp. Arch.*, pages 46–53. IEEE, Jun 1989.

[8] K. R. Traub. *Implementation of Non-Strict Functional Programming Languages.* Pitman Publishing, London, 1991. Also published by MIT Press, Cambridge MA.

[2] There is a way to designate one thread as the "left" operand and the other as the "right," regardless of the order in which they happen to attempt the *join*.

[3] Because Monsoon is pipelined, at any given time there may be up to eight independent threads executing in different stages. The number of threads queued up for eventual execution may be as large as 32,000.

DEVELOPING DATAFLOW ALGORITHMS[1]

Robert E. Hiromoto
Computer Research & Applications
Los Alamos National Laboratory
Los Alamos, NM 87545 U.S.A.

AND

Anton P.W. Bohm
Computer Science Department
Colorado State University
Fort Collins, CO 80523 U.S.A.

INTRODUCTION

The design and validation of parallel algorithms is a rewarding and satisfying experience once the implementation has been completed and debugged. It is this latter task which can be extremely frustrating when dealing with a general purpose multiple instruction multiple data (MIMD) computer system. Errors in expressing parallel constructs give rise to unpredictable execution behavior, affecting both the resulting answer and the sanity of the programmer.

The notion of a high-level parallel programming language that insulates the programmer from the perils of asynchronous bugs and booby traps has been the goal of many researchers in the functional language community. In the last few years, significant progress has been made in this area. Language and compilation research has resulted in several very powerful, inherently parallel programming languages. Notable among these is Id, the work of Prof. Arvind's Computation Structures Group at MIT. Id has a functional and deterministic subset, yet is a completely general purpose language supporting synchronizing data structures, and side-effects. Compilation research is also being carried out at Yale University, Chalmers University of Technology, and the functional programming group at the University of Glasgow.

Up until now few dataflow computer systems have been developed for wide use. One of the first was designed and built at the University of Manchester. This project resulted in identifying many important architectural issues in the design of support hardware for the dataflow execution model. The Manchester group used the purely functional, strict, single assignment language SISAL for writing their applications and produced a compiler generating highly efficient dataflow code for this language. A similar dataflow project was initiated at the Electro-Technical Laboratory (ETL) in Tsukuba, Japan. Here a large dataflow system with 128 processing elements was designed and built. Unfortunately the lack of programming software has prevented the system from being fully tested on substantial application codes. Still from these and other experiences, a multithreaded execution model with in the frame work of dataflow has emerged and has many researchers very hopeful of its success. Today several research groups are seriously involved with building prototypes of these multithreaded architectures (e.g., Sandia National Laboratories' (Albuquerque, New Mexico) Epsilon-2, Motorola and MIT's Monsoon, IBM's Empires, and ETL's EM-4).

We are now at a very exciting moment when language, compiler technology, and hardware are quickly maturing and becoming readily available for use. This moment also offers us the important opportunity to critically assess the advantages claimed by functional language and dataflow advocates.

Our approach is to study the performance of a collection of numerical algorithms written in Id which is available to users of Motorola's dataflow machine Monsoon. We will study the dataflow performance of these implementations first under the parallel profiling simulator Id World, and second in comparison with actual dataflow execution on the Motorola Monsoon. This approach will allow us to follow the computational and structural details of the parallel algorithms as implemented on dataflow systems. When running our programs on the Id World simulator we will examine the behaviour of algorithms at dataflow graph level, where each instruction takes one timestep and data becomes available at the next. This implies that important machine level phenomena such as the effect that global communication time may have on the computation are not addressed. These phenomena will be addressed when we run our programs on the Monsoon hardware. Potential ramifications for compilation techniques, functional programming style, and program efficiency are significant to this study. In a later stage of our research we will compare the efficiency of Id programs to programs written in other languages. This comparison will be of a rather qualitative nature as there are too many degrees of freedom in a language implementation (on language, compiler, and target machine level) for a quantitative comparison to be of interest.

We begin our study by examining four routines that exhibit different computational characteristics. These routines and their corresponding characteristics are listed below:

(1) **Fast Fourier Transforms (FFT)**
- *computational parallelism*
- *data dependences between the butterfly shuffles*

(2) **Adaptive Quadratures**
- *dynamic unrolling of recursively adaptive grid refinements*

(3) **Eigenvalue/vector Solvers**
- *application of cyclic rotations (incremental array updates)*

(4) **Stochastic Simulations**
- *data accumulation*

Details of our implementation and performance analysis will be presented during the session's presentation. For economy, we chose to describe only one of the four routines that have been analyzed at this time.

FFT

A fast fourier transform (FFT) routine was written in Id by J. Michael A•hley, a former summer student at Los Alamos National Laboratory. The FFT exhibits both divide and conquer and loop parallelism. The relevant Id program segments are:

```
Def Main_fft Size_of_V =
{ C = { array (1, Size_of_V) | [i] = Cmplx (i * 1.0) 0.0
              || j <- 1 to Size_of_V } ;
In FFT C } ;

Def FFT V = { (_,Size_of_V) = bounds V ;
In {if (Size_of_V == 1) then V
   else
   {(Left_V, Right_V) = Shuffle V ; FFT_L = FFT Left_V ;
    FFT_R = FFT Right_V ; Mid   = fix (Size_of_V / 2) ;
    X    = Two_Pi / Size_of_V ;
    Coeff = { array (1, Mid)
         | [i] = Cmplx (cos (X * (i-1))) (-sin (X * (i-1)))
         || i <- 1 to Mid };
    Prod  = { array (1, Mid)  | [i] = Cmplx_Mul Coeff[i] FFT_R[i]
         || i <- 1 to Mid }; };
    In { array (1, Size_of_V)
      | [i] = Cmplx_Add FFT_L[i] Prod[i] || i <- 1 to Mid
      | [Mid+i] = Cmplx_Sub FFT_L[i] Prod[i] || i <- 1 to Mid } } };
```

[1] This work was done under the auspices of the U. S. Department of Energy contract No. W-7405-ENG-36.

```
Shuffle V = { (_, Size_of_V) = bounds V ; Mid = fix (Size_of_V / 2) ;
In (( array (1, Mid) | [i] = V[(i*2)-1] || i <- 1 to Mid };
   { array (1, Mid) | [i] = V[i*2] || i <- 1 to Mid }) } ;
```

Note that in FFT a recursive invocation of FFT is applied to Left_V and Right_V (the odd and even elements of V, respectively) until the butterfly shuffle on V has been completed. The array element data dependences occurring in the recombination of smaller results into larger ones are expressed in three array comprehensions defining the values of the arrays Coeff, Prod, and the result of FFT. Running the above program under the Id world simulator for a Size_of_V of 128 gives us the following parallelism profile:

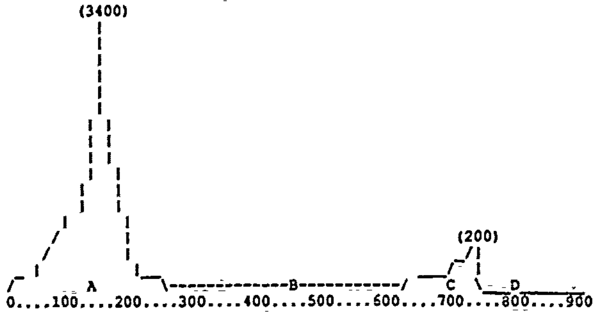


Fig. 1. Parallelism profile of FFT 128.

ANALYSIS

The simulator assumes that each instruction takes one timestep and that the results of an instruction execution are available the next timestep. This approximation leads to an idealised graph interpretation in which maximally parallel execution proceeds along a critical path via a sequence of indivisible timesteps. The graph in figure 1 plots the number of instructions executed at each timestep, and consequently pictures the ideal parallelism profile of FFT 128.

When studying Fig. 1, we observe the following. First there is explosive divide and conquer parallelism (A), followed by (B) a stretch of low parallelism. A second less significant burst of parallelism (C) follows which dies down to an almost sequential tail (D). For larger problems the two sequential stretches (B and D) are observed to dominate more and more. The parallelism profile drawn in Fig. 1 is very disappointing since the computational parallelism is known to be very large. To begin to understand this, we know that the FFT program takes O(log(n)) parallel steps to unfold all FFT and Shuffle functions. This accounts for the first burst (A) of the divide and conquer parallelism. Once the functions have been unfolded, the loops (array comprehensions) dictates the parallelism and consequently the speed of the computation. In the first instance the dominant loop is the array comprehension in the main function creating the original function values to be transformed.

Since a considerable amount of work may go into the activation of a loop-body, a loop analysis is performed on two simple loops:

```
Def WW n m = { s = 0;  r = 0;
In {while (s < n) do next s = s + 1; next r = r + W m; finally r} };
```

where

```
Def W n = { s = 0;
In {while (s < n) do next s = s + 1; finally s} };
```

Through an analysis of these two loops, it is found that WW (a doubly nested loop) requires five (5) steps on the critical path to instantiate each inner loop-body, that is, every five parallel steps a new inner loop is spawned off. As the inner loop bodies are skewed on top of each other, the number of them that run in parallel is equal to the critical path required to execute an inner loop divided by the rate at which the loop can spawn off the next inner loop. Figure 2 sketches a parallelism profile for WW 64 20:

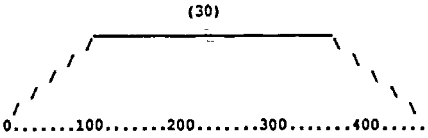


Fig. 2. Parallelism Profile for a nested loop.

Clearly, the loop rate plays an important role in the parallelism of a program. Phase B in Fig. 1 show this loop behavior where the dominant loop is the array comprehension in the main function creating the original function values to be transformed. Once every 5 parallel steps an array element is created and sent through the log(n) stages of the FFT.

The completion of the FFT is dependent on the loop rate and the creation of the last two elements of the original array. When the last two elements go into the FFT (phase (C) in Fig. 1), the remaining stages of the computations can be done with divide and conquer parallelism.

The last sequential tail (phase (D) figure 1) is caused by the array comprehension in FFT that generates the resulting array.

SOLUTION

The solution to this problem is to unroll loops fast enough so that they don't cause unnecessary delay. This has been recognized by a number of dataflow teams, who invented special instructions (very similar to vector instructions) to rapidly create parallel workload especially in loops (iterative instructions, repeat mechanism). Id does not have this type of instructions, but it is still possible, although at a consirably higher cost, to create array elements at a higher rate in a nested loop. Take the following array comprehensions:

```
Def A n = { array (1, n) | [i] = i || i <- 1 to n};

Def Ab n = { array (1, n) | [i] = i || j <- 1 to n by 16 & i <- j to j+15};
```

Some statistics:

A	n	S1	Sinf	Ab	n	S1	Sinf
	16	223	101		16	277	118
	32	399	181		32	504	123
	64	751	341		64	958	133
	256	2863	1301		256	3682	198

In the above table, S1 stands for the number of instructions executed and Sinf stands for the total critical pathlength. Where array elements are created at the loop rate (5) in A, they are created at a rate of 16 elements in 5 parallel steps in Ab (i.e., an element production rate of 16/5). This comes at the cost of higher S1 figures. This trick can be employed to make all loops in the FFT program go at a higher element production rate. Applying this idea to Main_fft, we have:

```
Def Main_fft Size_of_V =
{ C = { array (1, Size_of_V) | [i] = Cmplx (i * 1.0) 0.0
                || j <- 1 to size by 16 & i <- j to j+15 } ;
In FFT C } ;
```

To apply this technique to FFT, requires keeping track of the shrinking vector lengths. The application of this technique to the array Coeff in FFT results in:

```
if Mid > 16 then
{ Coeff = { array(1, Mid)
      | [i] = Cmplx (cos (X * (i-1))) (-sin (X * (i-1)))
      || j <- 1 to Mid by 16 & i <- j to j+15} ;
  In { array (1, Size_of_V)
      | [i]  = Add_c FFT_L[i] Prod[i]
      || j <- 1 to Mid by 16 & i <- j to j+15 } }
else
{ Coeff   = { array(1, Mid)
      | [i] = Cmplx (cos (X * (i-1))) (-sin (X * (i-1)))
      || i <- 1 to Mid } ;
In { array (1, Size_of_V)
      | [i]  = Add_c FFt_L[i] Prod[i] || i <- 1 to Mid } } }
```

Its parallelism profile is shown in Fig. 3.

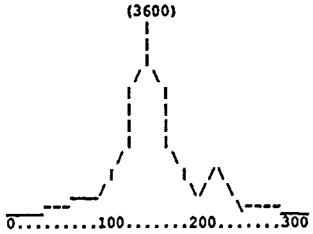


Fig. 3. Parallelism of unrolled loops in FFT-128.

The critical pathlength is about 30 percent of that of the original program, and the two ugly sequential stretches B and D have disappeared. Now the parallelism of the program is satisfactory.

It turns out that the array comprehensions are rather inefficiently implemented in Id. In the talk we will address the efficiency issue in some detail.

CONSTRUCTIVE ALGORITHMS AND PRUNING: IMPROVING THE MULTI-LAYER PERCEPTRON

Mike Wynne-Jones

Research Initiative in Pattern Recognition,* RSRE, St. Andrews Road, Malvern,
WR14 3PS, UK. Phone: (+44/0)684 894344, fax: (+44/0)684 894540
E-mail: mikewj@uk.mod.hermes

Abstract

A number of techniques have emerged recently, which attempt to improve on the multi-layer perceptron training algorithm [1], by changing the network architecture as training proceeds. These techniques include pruning useless or unnecessary nodes or weights, and adding extra nodes as required. The advantages to be gained are smaller networks, faster training times on serial computers, and increased generalization ability, with a consequent immunity to noise. In addition, it is frequently much easier to interpret what the network is doing. One can then begin to draw analogies with other pattern classifying techniques such as decision trees and expert systems. We review these techniques, suggesting the classes of problem to which they are most applicable, and indicate possible future work in this direction.

1 Introduction.

Multi layer perceptron networks are well established as a standard neural network technique for pattern recognition tasks. They are hampered, however, by a number of problems and arbitrary parameters which make them much less dependable and predictable than they could otherwise be. The main problem is that there is no way of determining in advance how many units there should be in the hidden layer. If there are too few, the network may not learn at all, while too many hidden nodes lead to over-learning of individual samples in the training data, at the expense of forming an optimal model of the data distributions. This leads to an inability to *generalise*, so that previously unseen data are labelled according to the nearest training sample, rather than in accordance with a good model of the problem. Despite numerous analytical and heuristic attempts to determine this number, [2,3] no general and reliable method has emerged until recently.

Recent developments fall into two catagories. *Pruning algorithms* build and train a large network, and then remove nodes which contribute little to the network's operation, while *constructive algorithms* attempt to form an approximate solution using a small network, and then add further nodes to increase the precision as required. It has been demonstrated [4] that a larger network is generally required to learn a classification task, than is needed simply to implement a known solution using predetermined network weignts. To this end, an ideal algorithm for determining network architecture might add nodes during early training, and apply pruning selectively as a solution is approached. The need for additional nodes to enable the learning of a solution arises because a network of the minimal size to solve a problem is likely to get stuck in one of many locally optimal solutions in the training phase. In constructive algorithms we hope that the local optima in which we find ourselves will represent a reasonably good approximation to the true solution, and we escape it by the addition of more free parameters to the network. To work well, the new parameters must be added with predetermined values so that the solution of the new network is at least no worse than that of its parent. By pruning weights (or nodes) which do not contribute significantly to the classification or mapping task, we hope to reduce to a minimum the number of degrees of freedom used by the network to implement the solution, thereby ensuring we have a simple model of the system. One way of making this obvious implementation of Occam's Razor, is to minimise the total *activation* in the hidden layer, which encourages the nodes there to act orthogonally. This is the same behaviour for which we aim in inductive inference, ie. in decision trees and automatically generated expert systems. It may be possible to extract information directly from the weights of such a network, allowing us to determine the problem-solving rules that have been learned. Naturally the reduction of the number of nodes to a minimum detracts from the fault tolerance usually associated with neural networks, and so some thought is needed before such minimalist techniques are used for a particular application.

2 The MLP training algorithm.

We briefly summarise the training algorithm described in [1]. In the notation, o_{pj} denotes the output of unit j, upon presentation of pattern p. Unit j is connected to unit i by a synapse or weight of strength w_{ij}, and computes

$$o_{pj} = f(\sum_i w_{ij} o_{pi}) \tag{1}$$

where f is a differentiable, non-linear function. The desired output for output unit j on presentation of pattern p is denoted t_{pj}, and the global error of the network after the presentation of some training patterns is defined as

$$E = \frac{1}{2} \sum_p \sum_j (o_{pj} - t_{pj})^2 \tag{2}$$

where the inner sum is over neurons considered to be output nodes, and the outer sum is over the patterns that have been presented.

Optimization is achieved usually by gradient descent, Newton's method, or conjugate gradients,[5,6] all of which use the derivatives of the network error with respect to the parameters to be optimised. We define an error measure in terms of the activations of output units, which can be differentiated; the term back-propagation refers to the repeated calculation of the corresponding derivatives for the previous layer. The chain rule for partial derivatives leads to the generalised delta rule,[1] which indicates how these quantities and the appropriate weight updates are to be calculated. Minimization of the sum squared error with respect to the free parameters in the network leads the network to approximate the Bayes discriminant vector, the probability of a class given the input to the network. [31]

3 Constructive Algorithms.

Early constructive algorithms built multi-layer feed-forward networks of perceptron units,[7] which could be applied to problems involving binary input patterns. Convergence of such a network is guaranteed if the training data is linearly separable.[8] The Pocket Algorithm [9] is an extension of the Perceptron Learning Rule,[10] which allows approximate solutions to be found for non linearly separable problems, by holding 'in one's pocket' the best set of weights found so far. For binary patterns, it is always possible to cut off a corner of the binary hypercube with a plane, thereby ensuring that we can learn a binary function of binary patterns by repeatedly adding perceptrons to a network. These techniques have been implemented in the *Tiling algorithm*,[11] although this been superseded by the *Upstart Algorithm*,[12,13] which is more efficient in terms of the number of nodes created. They build tree-like networks with finer resolution of the input space at the leaves of the tree, and do not usually include a stopping criterion to halt the addition of new nodes or layers. This means that every sample in

*Employed by THORN EMI Central Research Laboratories the training set is learned, but has strong repercussions if the training data is incomplete, has noise, or is derived from a classification problem where the class distributions overlap. These networks are good for learning completely a logical data set, but cannot form a useful model of many statistical distributions.

Later methods [14-18,34] apply to non-binary functions of non-binary inputs. They usually build a single hidden layer, which has an advantage over the 'deep network' methods that the propagation time for data from the input to the output is shorter, and constant. This is particularly important for real time signal processing applications, but not usually in simulated networks used for classification. They do not guarantee to learn every sample in the training set, but are more likely than the earlier algorithms to converge to solutions with good generalization ability for statistical problems.

3.1 Tiling Algorithm and Upstart Algorithm.

The Tiling Algorithm builds a layered network, with each layer approximating the solution of a binary function of classification problem more closely than the previous one. A *master unit* in each layer learns at least one new pattern, and the *ancillary units* ensure that no previously learned patterns are lost. The algorithm creates only as many nodes as it needs to build a solution, and it has been demonstrated to have good generalization capabilities. On the down side it requires an excessively large number of nodes, as most ancillary units just duplicate the action of those in the previous layer. This problem is avoided in later methods (Upstart, Cascade Correlation), by allowing connections to span many layers. A recent method which has grown out of the Tiling Algorithm is the *Neural Tree*,[35] which makes repeated use of the Pocket Algorithm to divide the input space into partitions for classification, and builds a decision tree[36] as it does so. No results have been published to date, but the method looks very promising.

The Upstart Algorithm builds a binary tree of nodes. For each node, one subtree corrects all errors where a one is expected, and the other corrects all errors where a zero is expected. Each subnode is guaranteed to classify at least one of its targets correctly, and so convergence is guaranteed for the problem. The number of nodes grows linearly with the number of training patterns for the theoretically hardest problem, a random boolean function, and this is better than the node growth rate of the Tiling Algorithm. Training can be speeded up if each subtree is trained using only the patterns that have not already been learned elsewhere in the tree. Extensions are possible whereby the trained network is mapped directly onto a feed forward network with only one hidden layer. The Upstart Algorithm is easily extended to classifiers with several possible output classes, while preserving many of its advantages.

3.2 Dynamic Node Creation in Standard MLPs

Ash [15] describes a system where the error rate is analysed as training proceeds, and a new hidden node is added whenever the error is no longer decreasing significantly, but is not acceptably low. The new nodes are introduced with small random weights, and the results are encouraging when compared with the training times required by standard networks of fixed size. The scheme includes an analysis of when new nodes should be added, based on the assumption that the error rate decreases exponentially as training proceeds, but the author feels that a better way is needed for determining the initial weight values of the new nodes, as small random weights do not guarantee that the larger network is at least as good as the old one.

A similar scheme [16,31] adds new nodes to the hidden layer, but speeds up training by freezing the weights of the previous ones. Back propagation's property of attempting to force each hidden node to account individually for *all* the error in the output layer means that the newly introduced nodes learn the errors of the previous network with frozen weights. Interestingly, this feature is usually considered a disadvantage, as the use of 'teamwork' amongst nodes might be expected to find a solution more quickly.

3.3 Cascade Correlation

In this algorithm, which applies to real-valued inputs and outputs, hidden units are added with inputs from all previous input and hidden units.[14] The weights to all previous units are frozen, and the new unit learns a mapping which has the best possible correlation with the errors of the previous network. The weights to the new unit are then frozen, and the process continues until there are no more errors. This method is not limited to binary classifications, although published results only cover these problems. There is no back-propagation, with the result that the networks train quickly; the speed compares very favourably with both backprop and 'quickprop' [19,20] when learning to separate two interlocked spirals. The problem of learning individual samples from overlapping distributions might be addressed by continuing the development of a network until the error figure found by correlation decreases to the value expected from prior knowledge of the problem. This could be a noise model, or a measure of the degree to which the data distributions are known to overlap. Since Cascade Correlation builds tall networks, with many layers and connectivity from each layer to all earlier layers, it has the advantage of enhanced *feature detection* over the networks described in the previous section. On the other hand, the many-layered architecture leads to a variable delay between application of an input and the appearance of the result.

3.4 Meiosis Networks.

Meiosis networks [17] arise from an approach to multi layer perceptrons which combines the usual gradient descent optimisation with a stochastic search. This has the advantage that it is possible for the network to escape from a local minimum by a random perturbation in the weight space, with a probability which decreases as the network approaches a good solution. This is implemented using stochastic weights, sampled from gaussian distributions each time a weight value is required. The learning algorithm updates the mean and the standard deviation of this distribution. Thus there are three learning rules. the mean is updated according to the normal learning rule described in §2. The standard deviation is increased at each update to reflect the uncertainty indicated by a large update, and decays with time, allowing the weights eventually to become deterministic.

Meiosis is the process of one node splitting to create two new ones. The composite variance of input and output weights is computed for each node in the hidden layer, and the split occurs for any node whose composite variance, that is the standard deviation relative to the mean, is greater than 100%.

$$\frac{\sum_i \sigma_{ij}}{\sum_i \mu_{ij}} > 1 \text{ and } \frac{\sum_k \sigma_{jk}}{\sum_j \mu_{jk}} > 1. \tag{3}$$

Each new weight has a mean which is a jittered copy of the original, and each has half the variance of the old weight distribution. This kind of splitting policy has the advantage that it does not converge to a complete fit of the training data, and consequently the resulting networks are likely to exhibit better generalization than the ones produced by the Pocket, Tiling and Upstart algorithms. It has the additional advantage that the decision on whether to split is made using only locally measured parameters.

3.5 Summary: What to do next with constructive algorithms

The author feels that constructive algorithms lack a good mechanism for determining the weights of a newly added node, although Meiosis is good in the context of stochastic weights. Accepting the idea of adding new nodes by splitting an old one in two, we also require a good measure of the degree to which a given node requires splitting. In our own work,[18] we propose a new split mechanism, and build on earlier ideas from *pruning*, to obtain an ordered list of nodes with the most prunable at one end, and the most splittable at the other.

4 Pruning

Pruning has been carried out on networks in three distinct ways. The first is a heuristic approach, based on identifying which nodes and weights contribute little to the mapping. After these have been removed, additional training leads to a network which is better than the original.[4] An alternative technique is to include terms in the error function, so that the weights tend to zero under certain circumstances. Zero weights can be removed without degrading the performance of the network. Finally, if we define the sensitivity of the global network error to the removal of a weight or node, and evaluate it for each such parameter in the network, we can then remove the weights or nodes to which the global error is least sensitive. The sensitivity measurement does not interfere with training, and involves only a small amount of extra computational effort. It is also well matched for implementation alongside the node-adding scheme described in §3.5.

4.1 Heuristic Pruning

Sietsma and Dow [4] have described a pruning scheme based on analysing automatically the activations of nodes, in response to different patterns in the training data. Their pruning takes place in three different ways:

- Pruning of units that do nothing, or duplicate the action of other units. A unit is redundant if it has the same output for all patterns in the training data, or duplicates the action of another node.
- Pruning of units that provide unnecessary information. If a node makes a distinction between patterns that are later recombined, then the node can be deleted.
- Pruning out an entire network layer. A network with many layers may well have one or more of them redundant, especially after the earlier stages of pruning have been carried out. Indeed, it has been shown [21,33] that one hidden layer is theoretically sufficient to implement any problem, although this may require more nodes than a multi layer network.

4.2 Pruning which is inherent in the learning algorithm

This lower-level aproach is more appealing, as the constraints on structure for the network come from the network itself, rather than from a global monitoring system as described in §4.1. On the other hand, systems have been proposed where processes such as adding nodes and pruning would be carried out by exactly this type of global control process, which could itself be a rule based system or a neural network.

4.2.1 Minimising a Biased Cost Function

In standard learning techniques we optimise a network in terms of a cost function which describes the goodness of fit of the model stored in the network, to the data representing the problem to be learned. A second term can be added to the cost function (eq.4) incorporating prior knowledge of the problem, such as the kind of architecture which is likely to be most useful for the problem, constraints on weight values, or constraints on the function implemented by the network. A constraint on the function implemented might be that the network should not learn to classify individual samples from overlapping or noisy distributions; elimination of such unlikely solutions leads to the extra term being referred to as a *bias or regularising term* in the field of non-parametric statistics.

$$O = E + B \tag{4}$$

In relation to pruning networks, or the production of minimal architectures, the constraints we wish to apply by means of the bias term will be such that they minimise the number of active nodes or weights in the network. These can be removed from the network before training is continued.

Rumelhart's pruning mechanism (unpublished) involves a bias term of

$$B = \sum_i \sum_j w_{ij}^2 \tag{5}$$

giving a modified weight update rule

$$w_{ij}(n+1) = -\epsilon\frac{\partial E}{\partial w_{ij}}(n) + \beta w_{ij}(n). \tag{6}$$

This causes the weights to decay exponentially towards zero, and this aproach was successfully used by Hinton in some experiments on analysing the use of hidden layers and a representational bottleneck to encourage good generalisation.[32] Weights which come sufficiently close to zero in the weight decay scheme can be pruned out. This work was taken a step further by Hanson and Pratt [22], and Rumelhart, with the aim that high and low weights should decay strongly, while mid-range weights are left unaffected. High weights are discouraged for reasons of smoothness, while low weights can be pruned. These experiments showed that different bias terms are indeed useful for finding minimal architectures, but the bias term could not easily be used in conjunction with the momentum term used traditionally for avoiding local minima. Perhaps the biased cost function could be combined with Hanson's stochastic search MLP so that the traditional momentum term would no longer be needed.

Biased cost functions have been used in other applications too, first for pruning *nodes* by including a total *node relevance* term in the cost function.[23] The relevance term is defined as a product of functions of the weights into and out of a node, and minimising the cost function then minimises the total number of relevant nodes. This kind of *node decay* (cf. weight decay, above) has not been analysed in detail, but gives promising initial results. A second example encourages the *optimal-use of hidden units*, [29] by including the hidden layer activations in the cost function to be minimised, and hence forcing the units to act orthogonally. This attempts to avoid unnecessary units right from the start of training, rather than allowing them to develop and eliminating them later. This technique could prove useful in the context of integrating neural networks with expert systems or decision trees, since these systems attempt to invoke independent rules whenever possible.

The initial aim of Rumelhart's use of the biased cost function was to avoid large weights, and consequently to ensure smoothness of the mapping implemented by a network. A sigmoid transfer function is linear for small input values, but approximates to a step function for larger values (and hence for small weights;) step functions are necessary to classify individual samples from a statistical distribution and we have already been emphasised in §3 that this should be avoided if at all possible.

An alternative approach to discouraging the fitting of individual samples. again emphasising the power of the biased cost function, is to include a term in the cost function representing the curvature of the mapping, averaged over the input domain.[30] The elimination of high curvature eliminates sharp transitions, and has a similar effect to the bias term in equation 5. In addition, if the data distributions are smooth, and the curvature of the mapping is minimised, the performance is likely to be improved for the classification of previously unseen patterns. Minimal networks implementing low curvature mappings would also be more likely to perform usefully in extrapolation, that is for the classification of patterns outside the input domain represented by the training data.

4.2.2 Pruning according to an ordered list of node or weight relevances

Mozer and Smolensky [24] opt for an 'all or none' approach to pruning, rather than gradual pruning by means of weight decay. They investigate the relevance of a unit in the network, defined as

$$\rho_i = E_{\text{without unit } i} - E_{\text{with unit } i} \tag{7}$$

so that the relevance of a unit depends on how much the network global error will increase if the node is removed. Pruning could then be ap plied for the least relevant units. The relevance measure is determined by the shape of the error surface around the minimum to which the network has been trained. This is best characterised for a parabolic minimum by the first and second derivatives according to the Taylor series, namely

$$E(a) = E(a_0) + \left.\frac{\partial E(a)}{\partial a}\right|_{a=a_0} (a - a_0) + \frac{1}{2} \left.\frac{\partial^2 E(a)}{\partial a^2}\right|_{a=a_0} (a - a_0)^2 \qquad (8)$$

although Mozer and Smolensky propose to approximate it by carrying out a single pass of back propagation with a linear error function, $E = | o_{pi} - t_{pi} |$, so that the gradient is non zero right down to the error minimum. The question of what parameter of a node should be used as the measure against which the shape of the error surfaced is measured, is resolved by introducing the *attentional strength*, α of a unit, where the output of the unit is now $o_j = f(\sum_i w_{ij}\alpha_i o_i)$ and the relevance can be expressed as $\rho_i = E_{(\alpha_i=0)} - E_{(\alpha_i=1)}$. Relevance is determined by the shape of the error surface around the value $\alpha = 1$. This pruning technique led to networks from which rules could be extracted quite easily for low dimensional boolean problems, and the relevance measures facilitated an evaluation of which rules were the most frequently invoked in a given problem, ie. which splits in the feature space are the rules, and which are the exceptions. Le Cun, Denker and co-workers [25,26] have carried out similar experiments for pruning weights, but used the second derivative term from the Taylor series, instead of the modulus approximation. The second derivatives are found by back-propagation[25,26], (or by measurement[27]) and the results are very promising. The number of free parameters in the networks used to implement handwritten digit recognition [26] was reduced by a factor of four. There are no results available of comparisons between this pruning methods and those discussed earlier, but this gives by far the most convincing way of identifying which nodes are to be pruned. The idea of an ordered list of sensitivities of the network global error to each weight was used also by Karnin.[28] Ideally the sensitivity would be determined by integration over the entire weight space, but since this is not possible, (cf. training by exhaustive search), it is integrated just along the training path. It seems likely that the sensitivity found in this way would depend most strongly on the starting weights (ie. random) and not be as good as the measures described above.

5 Discussion

We have discussed a number of techniques which aim to build a neural network whose size, and hence whose number of internal parameters, is optimal for the modelling of a given problem. While some problems are most efficiently modelled by certain types of network, the multi layer perceptron family are a good general learning tool for a wide range of modelling applications. Since we do not know in advance what size to use, and because we usually need a larger network to learn a mapping than simply to implement a known solution, it is sensible to allow a small network to grow during early training, until a reasonable solution is found, and then to optimise this solution during later training, to give a small, fast and efficient network which is an accurate system model.

There are a variety of pruning mechanisms, and constructive algorithms are gradually appearing which can add new nodes at suitable times. The weights for the new nodes can be predetermined to ensure firstly that the new network is no worse than the old one, and secondly that it can be expected to find a better solution with further training.

We have seen that a measure of the sensitivity of the network to the presence of weight or node forms a good criterion for determining whether it can be removed, and that at the other end of the scale, if applied to nodes, the same mesasure can be used as a criterion for splitting a node in two.

Of the current constructive algorithms, *Upstart* appears to be the best for binary mappings, while the best for real valued mappings is *Cascade Correlation* Both of these have the problem of long propagation delays from the network inputs to the outputs, although this is solved for *Upstart* by transforming the trained network. The best criterion for pruning appears to be the measure of the sensitivity of the network error to the presence of a parameter, and it is hoped that this will very soon be used as the basis of an integrated system for building networks, incorporating construction in early training and pruning as a solution is reached, enabling an optimal architecture to be found.

6 References

1 DE Rumelhart, GE Hinton & RJ Williams Rumelhart & McClelland (eds.), 'Parallel Distributed Processing' vol 1, ch 8 Bradford Books / MIT Press, 1985.

2 ID Longstaff & JF Cross. Pattern Recognition Letters, 5 315, 1987.

3 M Gutierrez, J Wang & RO Grondin. First IEE International Confercence on Neural Networks, October 1989, pp 120-124.

4 J Sietsma & RJF Dow IJCNN, 1988, 1 pp 325-333

5 DG Luenberger. 'Linear & Non-Linear Programming' Reading, MA: Addison-Wesley, 1984.

6 AR Webb, D Lowe, & MD Bedworth, July 1988. RSRE memo 4157, Royal Signals & Radar Establishment, Malvern, WR14 3PS, UK

7 F Rosenblatt. Psychological review, 65 368-408, 1958

8 HD Block. Reviews of Modern Physics, 34 123-135, 1962

9 SI Gallant. 'Optimal Linear Discriminants' IEEE Proceedings of the Eighth Conference on Pattern Recognition, Paris, 1986.

10 F Rosenblatt. 'Principles of Neurodynamics'. Spartan Books, New York, 1962.

11 M Mezard & JP Nadal. J. Phys. A. Math. Gen. 22 2191-2203, 1989.

12 M Frean, 1990. Short Paths and Small Nets. Optimising Neural Computation.' Ph.D. thesis, University of Edinburgh, Department of Physics, Edinburgh, EH9 3JY, UK.

13 M Frean. Neural Computation 2, 2, 198-209, 1990

14 SE Fahlman & C Lebière. In 'Advances in Neural Information Processing Systems 2', ed. DS Tourzetsky, Morgan Kaufmann, 1990, pp.524-532.

15 I Ash, 1989. ICS report 8901, Institute of Cognitive Science, University of California, San Diego, La Jolla, California 92093.

16 N Thacker, AIVRU, Sheffield University, UK, personal communication

17 SJ Hanson. In 'Advances in Neural Information Processing Systems 2', ed. DS Tourzetsky, Morgan Kaufmann, 1990, pp 533-541

18 M Wynne-Jones, 1990. RIPRREP/1000/82/90, Research Initiative in Pattern Recognition, RSRE, Malvern, WR14 3PS, UK.

19 SE Fahlman. 'Faster-Learning Variations on Back-Propagation: An Empirical Study' In 'Proceedings of the 1988 Connectionist Models Summer School', Morgan Kaufmann, 1988.

20 SE Fahlman, 1988. Technical Report CMU-CS-88-162, Carnegie Mellon University.

21 K Funahashi. Neural Networks 2 183-192, 1989.

22 SJ Hanson & LY Pratt. In 'Advances in Neural Information Processing Systems', ed: DS Tourzetsky, Morgan Kaufmann, 1989, 107-115.

23 J Chuanyi, RR Snapp & D Psaltis. Neural Computation 2, 188-197, 1990.

24 MC Mozer & P Smolensky. In 'Advances in Neural information Processing Systems', ed. DS Tourzetsky, Morgan Kaufmann, 1989, pp.107-115.

25 Y Le Cun, JS Denker & SA Solla. In 'Advances in Neural Information Processing Systems', ed. DS Tourzetsky, Morgan Kaufmann, 1990, pp.598-605.

26 Y Le Cun, B Bozer, JS Denker, D Henderson, RE Howard, W Hubbard & LD Jackel. Neural Computation 1 4, 1990.

27 WH Press, BP Flannery SA Teukolsky, & WT Vetterling. 'Numerical Recipes in C. The Art of Scientific Computing' Cambridge University Press, 1986

28 ED Karnin. IEEE Transactions on Neural Networks, 1 2, 1990.

29 Y Chauvin. In 'Advances in Neural Information Processing Systems', ed. DS Tourzetsky, Morgan Kaufmann, 1989, pp.519-526.

30 CM Bishop 'Curvature-Driven Smoothing in Backpropagaton Neural Networks', INNC, Paris, 2, 749-752.

31 D Lowe & AR Webb. 'Optimised feature extraction and the Bayes decision in feed-forward classifier networks', IEEE Trans. Pattern Analysis & Machine Intelligence, 1991 (in press).

32 GE Hinton. In PARLE. Parallel Architectures & Languages for Europe, eds. G Goos & J Hartmanis, Springer Verlag, Berlin, pp.1-13.

33 K Hornik, M Stinchcombe & H White. Neural Networks 2 pp.359-366, 1989.

34 AN Refenes & S Vithlani. 'Constructive learning by Specialisation', to appear in Proc. ICANN, Helsinki, Finland, June 1991.

35 JA Sirat & JP Nadal. 'Neural Trees. A New Tool for Classification', submitted to Network April 1990.

36 L Breiman et al. Classification and Regression Trees, Wadsworth, 1984.

Neural Networks as Petri Nets

Hans Nieters, GMD, 5205 St. Augustin, Postfach 1240, Germany
Email: nit@gmdzi.gmd.de

Abstract

Relative few papers on the relationship between Neural networks (NN) and Petri nets (PN) are known (e.g. [1],[5]). Other groups try different approaches: M.J.Murre (Leiden University, Netherland) and K.Lautenbach (University Koblenz-Lindlau, Germany). Our approach is to use Predicate Transition systems (PrT systems). The result is, that for nonrecurrent NN's a behaviorally equivalent PN can be given. The proposed technique of transforming NN's (with and without backpropagation) into PN's is demonstrated by examples. We show some differences between both modeling techniques and also, that both could perhaps gain benefits from each other, if they were put on a common basis.

NN as PrT systems

We follow about the NN definitions given in [4]. Predicate Transition systems (PrT systems) are a widely used high level class of PN. Details about PrT systems can be found in [3],[7]. The representation of a NN (without backprop.) into a PrT system is done by associating to each unit a transition and to each link a place. Weights occur in the 'equations' associated to transitions, where also the computation of activation values is performed. Example: The NN for the XOR-problem is shown in Fig 1, the corresponding PrT system in Fig. 2.

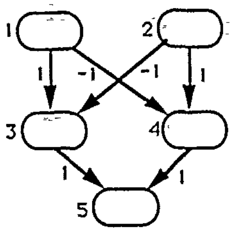


Figure 1: A NN for the XOR problem.

The input marking corresponding to the input pair (x,y) is placed in the input places i1, i2 such that $M_0(i1) = x, M_0(i2) = y$, where M_0 initial marking. The corresponding PrT system is 'behaviorally equivalent' to the NN, in the sense that for each input given to the input units of the NN, the marking of the output place(s) becomes (after firing of the transitions) identical to the output values of the output units (after updating the corresponding units).

Lets explain the PrT system of Fig. 2 in detail. The transitions correspond one-one to the units of the NN (1 to 5). For input units we add input places (i1, i2), for output units an output place (o5). For each link between two units we generate a place and connect that with the units. The arcs are labeled i.g. with formal sums of tuples $< t_{11}, \dots, t_{1n} > + \dots + < t_{m1}, \dots, t_{mn} >$, where the t_{ij} are either variable symbols (starting with capital letters) or terms over some set of operation symbols and variable symbols. In our example $m = n = 1$ and each t_{ij} is a variable symbol. The markings are taken from that set of formal sums too, but no variable symbol must occur in the t_{ij}. The labels of arcs in the neighborhood of a place must be of the same arity (n) as the marking of that place. The empty tuple $<>$ is allowed too in markings and arc labels. To each transition a conjunction of equations between two terms may be associated. The equations must all be satisfied for the transition being enabled.

In order to describe the dynamics of PrT systems assume as initial marking e.g. $M_0(i1) = 0, M_0(i1) = 1$. Transitions 1,2 are enabled now. Transition 1 is enabled for the substitution $\{0/I1, 0/A1\}$, transition 2 is enabled for the substitution $\{0/I1, 0/A1\}$. Under these substitutions the equations associated to the transitions become $0 = 0$ and $1 = 1$ resp. and hence are satisfied. Both transitions may fire concurrently, removing the tokens from their input places i1, i2 and adding tokens $< 0 >$ to their output places a31, a41 (resp. $< 1 >$ to a32, a42). Next, transitions 3,4 are enabled under the substitutions $\{0/A1, 1/A2, -1/Net3, 0/A3\}$ and $\{0/A1, 1/A2, 1/Net4, 1/A4\}$. After concurrent firing of these transitions, tokens are removed from places a31, a41, a32, a42 and tokens $< 0 >$ ($< 1 >$) added to a53 (a54). Last, transition 5 is enabled under substitution $\{0/A3, 1/A4, 1/Net5, 1/A5\}$ and results in marking $< 1 >$ on place o5, which is the result expected for the given input.

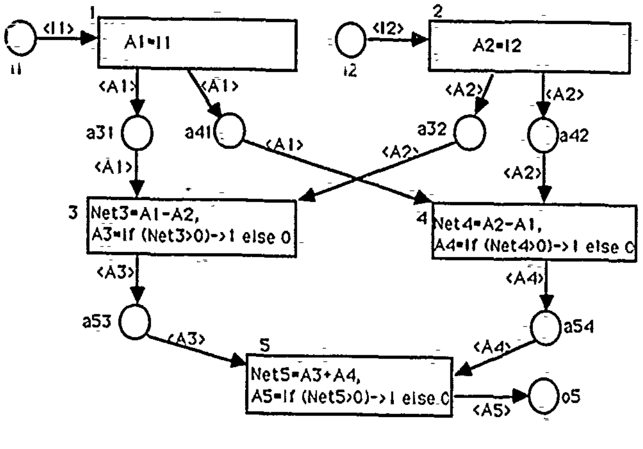


Figure 2 The PrT net for the XOR NN.

For NN without backpropagation that solution is straightforward. The update (firing) is maximally concurrent, we dont need any extension of classical PrT systems and the Prt net can be generated automatically from the NN (and this has in fact been done).

NN with backpropagation

The translation of a NN with backpropagation requires a single 'global' transition for setting the activation values (for input units), the target values (for output units), and for testing whether to stop updating or starting the next update with new input/target vector. We give now the PrT transition(s) for each (input, hidden, output) unit of a NN.

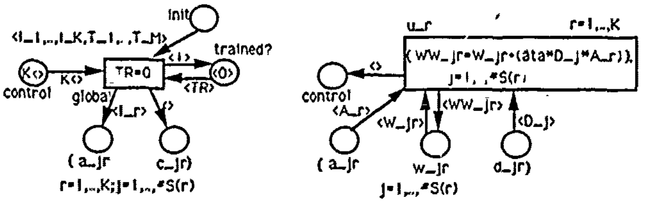


Figure 3: The global transition and the transition for an input unit with Bp.

Global' takes input from three global places 'init', 'control' and 'trained'. init contains the vector of input and trainings values for one computation cycle, including -1 values for the threshold input units. For example, in case of the XOR example above, the initial marking of init' could be $1000 < -1,0,0,-1,-1,0 > +1000 < -1,0,1,-1,-1,1 > +1000 < -1,1,0,-1,-1,1 > +1000 < -1,1,1,-1,-1,1 >$, where the second and third components of each tuple $< \dots >$ denote the input values, the last component the target value and the others the threshold input values. The global place control' is initially marked by $K <>$, where K is the number of input units and all of them are consumed when 'global' fires. The global place 'trained?' initially contains a $< 0 >$ token, which is altered to $< 1 >$ in case of firing, which may happen only if the equation 'Tr=0' is satisfied. I.e. global' fires exactly once for each computation cycle. The transition produces output tokens for each pair of places c_{jr}, a_{jr} where r runs over the set of input units and j over the set of successors S(r) of unit r. The arcs are labeled with a control token $<>$, resp. with $< I_r >$, where I_r denotes a variable symbol (r replaced by the same index as in its corresponding place).

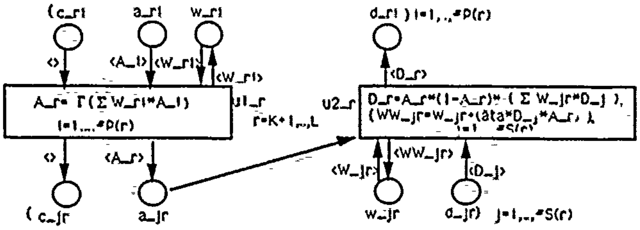


Figure 4: The PrT transitions for a hidden unit with backpropagation.

For each input unit r in 1,.., K a transition u_r is generated, which is part of the backward pass (cf. righthand side of Fig. 3). It takes its own activation value A_r (set by 'global' and used by its successor units), the old weights W_{jr} (connectors to its successors) and the error output D_j backpropagated from its successors (!). It produces new weights WW'_{jr}, changed by the transition equations, and a token on the 'control' place. The transition has an equation for each outgoing connection. All of them must be computed in order to produce the WW_{jr} values. (Symbols starting with capital letters denote variables). 'aeta' is a constant (0.5 for our example).

For each hidden unit two transitions must be generated (cf. Fig. 4). $u1_r$ for the forward pass, computing the new activation value, and $u2_r$ for the backward pass, computing the error signal D_r to its predecessors P(r) and then the new weights. The equation in $u1_r$ uses Γ, which is a sigmoidal squashing function. In $u2_r$ the denotation $j = 1,..,\#S(r)$ is used both as summation index in the first equation (the same as in $u1_r$) and as index for producing the rest of equations.

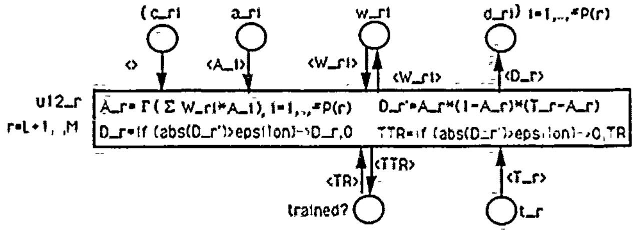

(c_rl
a_rl
w_rl
d_rl) l=1,..,#P(r)
<>
<A_l>
<W_rl>
<W_rl>
<D_r>
u12_r
r=L+1, ,M
A_r= Γ(Σ W_rl*A_l), l=1,..,#P(r)
D_r'=A_r*(1-A_r)*(T_r-A_r)
D_r=if (abs(D_r')>epsilon)->D_r,0
TTR=if (abs(D_r')>epsilon)->0,TR
<TTR>
<TR>
<T_r>
trained?
t_r

Figure 5: The PrT transitions for an output unit with backpropagation.

The schema for generating the transitions for output units should be clear now The initial value of 'trained?' is < 0 > and gets changed only if the stop criterion for that transition is satisfied The weight places W_{ij} are initially marked with the inital values. The result of generating the PrT system for the XOR example is left as an exercise to the interested reader.

PN + NN = Hybrid systems?

Some differences between PN and NN are. (1) Single type of nodes (NN) versus bipartite graph (PN). (2) Continuous activation values (NN) versus discrete (PN). (3) (Also negative) real values of weigths (NN) versus positive integer (PN). (4) Units are often updated synchronously (NN) but asynchronous transition firing (PN) (5) NN are deterministic, i.e. no decisions between alternatives are taken even in recurrent NN (not considered here. Hence the PN representation opens up the possibility to model alternatives additionally. (6) The embedding of propositional and first order predicate logic in PN theory has no equivalent in NN. The NN form of the logical 'or' is rather complicated. In PN we know that a propositional implication $a \wedge b \Rightarrow c \vee d$ is represented as a dead transition with a,b as input and c,d as output places. The representation of (even) propositional logic in NN is much more complicated. (7) PN are suited for formal analysis (reachability trees and invariant calculus), since PN are used to describe exact solutions for discrete systems, where the main problem is to guarantee that the system is live and save. Since NN are not used for designing systems like operating systems or protocols, NN units keep only a single actication value and distribute only that single value to all successor units. A restricted form of liveness condition occurs as inital condition in recurrent NN only. (8) Modifications of weights and hence 'learning' is usually not at all considered in PN. Exceptions are self-modifying nets [8],[2]. (9) Usual tools designed for editing and simulating PrT systems could in principle be used for the PrT form of NN, but the mere amount of connections up to 10^4 makes it necessary to use dedicated simulators instead of the general simulators for PrT systems at hand. E.g. the GRASPIN simulator ([7]), applied to the XOR-network with backpropagation, updates the network in 3 seconds, wheras a dedicated NN simulator manages at least 5 to 20 thousand updates per second.

PrT systems (like all other PN) can be 'folded'. The result is often a single transition with complex inscriptions. The result of folding the PrT system generated for the XOR-NN with Bp is shown in Fig. 6. The simulation of such transition is of course faster than simulating the original system, since the overhead for checking nonactivated transitions is greatly reduced. This holds at least for single processor machines.

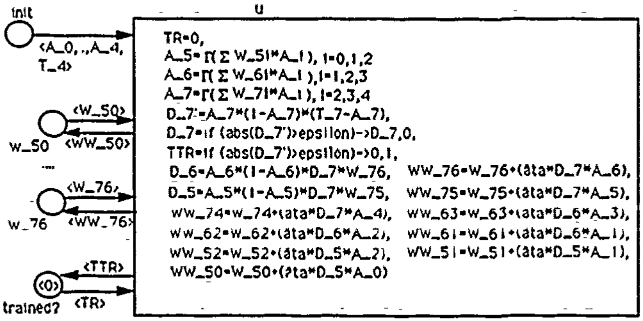

init
u
<A_0,..,A_4,
T_4>
TR=0,
A_5= Γ(Σ W_5l*A_l), l=0,1,2
A_6=Γ(Σ W_6l*A_l),l=1,2,3
A_7=Γ(Σ W_7l*A_l), l=2,3,4
D_7'=A_7*(1-A_7)*(T_7-A_7),
D_7=if (abs(D_7')>epsilon)->D_7,0,
TTR=if (abs(D_7')>epsilon)->0,1,
D_6=A_6*(1-A_6)*D_7*W_76,
D_5=A_5*(1-A_5)*D_7*W_75,
WW_74=W_74+(ata*D_7*A_4),
WW_62=W_62+(ata*D_6*A_2),
WW_52=W_52+(ata*D_5*A_2),
WW_50=W_50+(?ta*D_5*A_0)
WW_76=W_76+(ata*D_7*A_6),
WW_75=W_75+(ata*D_7*A_5),
WW_63=W_63+(ata*D_6*A_3),
WW_61=W_61+(ata*D_6*A_1),
WW_51=W_51+(ata*D_5*A_1),
<W_50>
<WW_50>
w_50
<W_76>
<WW_76>
w_76
<TTR>
<0>
trained?
<TR>

Figure 6. The PrT system for the XOR-NN (with Bp), completely folded.

Since the considered class of NN can always be folded in that way, the resulting transitions can be used as building blocks in hybrid systems, leaving the task of representing and analysing synchronization of complex systems to the very PN methods and tools. In the future we will consider recurrent NN and look for examples within which an answer can be established to the question how hybrid PN-NN models could look like and whether they can be combined in a fruitful way.

References

[1] J.Elz, *Petrinetze zu Simulation von Neuronenverschaltungen und einfachen Lernvorgängen.* (in German) Address of Author: J.Elz, Hans-Wilhelm-Hannen-Weg 9, 4600 Dortmund 50, FRG

[2] H.E.Fuss, *AFMG - ein asynchroner Flussmodellgenerator.* (in German) GMD Bericht Nr 100, 1975

[3] H.J.Genrich, *Predicate Transition Nets* . In: Petri Nets: Central Models and their Properties. LNCS 254, 1987

[4] A.Linden, *Untersuchung von Backpropagation in konnektionistischen Systemen.* (in German) Instituts-Bericht No. 80, Dept. of Computer Science, Bonn University Bonn. 1990

[5] M.K.Habib, R.W.Newcomb *Neuron Type Processor Modeling Using a Timed Petri Net.* IEEE Transactions on Neural Networks, Vol 1, No. 4, Dec 1990

[6] H.Nieters, *Graphical Simulation of Petri Nets in the GRASPIN environment.* ESPRIT Project 125, Technical Paper GMD 40/1, March 90

[7] H.Nieters, *Neural Networks as Predicate Transition Systems.* To appear as Arbeitspapiere der GMD

[8] R.Valk, *Self Modifying Nets. A natural Extension to Petri Nets.* in Ausiello/Bohm (ed:) LNCS 62, 1978, Springer Verlag.

The symmetric logarithmoid : an activation function for neurons

Abhay Bulsari, Alexander Medvedev, Björn Saxén and Henrik Saxén
Kemisk-tekniska fakulteten
Åbo Akademi, SF 20500 Turku / Åbo, Finland
E-mail : vt_ai@abo.fi

ABSTRACT. The symmetric logarithmoid provides a viable alternative to the sigmoid, while preserving many characteristics of the sigmoid. The sigmoid is very flat when the absolute value of its argument, $|a_i| > 10$. In other words, its derivative is extremely small, and has poor sensitivity to its argument. This is the root cause of the very slow rates of convergence during the training phases of neural networks, and relative insensitivity of the network to a fairly wide range of weights. The symmetric logarithmoid overcomes these limitations, despite perhaps creating some others of its own.

Feed-forward neural networks can be used as simulators trained from the gross observed behaviour of a system. This paper illustrates the applicability of the symmetric logarithmoid activation function in a feed-forward neural network for such training exemplified by a system identification problem of pressure drop in a rough pipe. The inputs to the network include viscosity, density of the liquid, diameter and roughness of the pipe, and the velocity. The outputs are the friction factor and the pressure drop per unit length. The networks were trained using the Levenberg-Marquardt method.

The symmetric logarithmoid is continuous, first order differentiable and a simple, monotonically increasing algebraic function. While minimising the error square sum for the outputs, convergence is generally fast compared to the sigmoidal activation function. Extremely large weights are not commonly generated by the training process, but is a usual feature with the sigmoids. The symmetric logarithmoid, evidently, does not mix well with other activation functions, especially the sigmoids.

1. Introduction

A lot of research has been done on feed-forward neural networks taking the sigmoid for granted. The sigmoid, however, has its limitations and its applicability is not universal. Sigmoids are meant for outputs contrained between 0 and 1, or −1 and 1. A linear mapping can extend this range. But many variables do not have such limits, and it is not always desirable to map them to a range of 0 to 1 since sensitivity can be lost in the process of mapping. Feed forward neural networks can be used for system identification of processes [1] , and one often comes across variables like temperature, pressure, viscosity, concentration, etc. which have no upper limits, although for a system under consideration, they may stay in a particular range. If the range is more than a couple of orders of magnitude, it is customary to deal with their logarithms in the mathematical models.

The sigmoid is very flat when the absolute value of its argument $|a_i| > 10$. In other words, its derivative is extremely small, and has poor sensitivity to its argument. This is the root cause of the very slow rates of convergence during the training phases of neural networks, and relative insensitivity of the network to a fairly wide range of weights.

The symmetric logarithmoid, given by the following equation overcomes these limitations, despite perhaps creating some others of its own.

$$x_i = \frac{a_i}{|a_i|} \ln\,(1 + \beta|a_i|)$$

The sigmoid and the symmetric logarithmoid can be considered to be in a continuum of activation functions. One extreme of the activation functions is the switch (the sign function), a network based on which cannot be trained by any of the optimisation methods meant for continuous functions. The sigmoid alleviates this problem by smoothening the switch near its discontinuity. The symmetric logarithmoid is continuous and first order differentiable. It is a monotonically increasing function with maximum sensitivity near zero and monotonically decreasing sensitivity away from zero, as with the sigmoid. However, the symmetric logarithmoid never becomes insensitive to the argument, and its output is not limited to between -1 and 1. Networks using this function are a bit easier to train, and the convergence is better. The other extreme of activation functions is the linear (identity) function, which finds limited use in our work for statistical purposes. This function, obviously poses no problems to the usual optimisation methods, and the Levenberg-Marquardt method, which we use, converges in a couple of iterations.

2. The flow in a rough pipe

To study the applicability of the symmetric logarithmoid, we considered a simple system of a flowing liquid in a rough pipe, under turbulent flow. The pressure drop in a pipe is of vital interest while sizing pumps or compressors, and for calculating flow rates. It is estimated by empirical correlations such as the one shown below [2] .

$$\frac{\Delta p}{L} = \frac{\zeta \rho w^2}{2d}$$

$$\zeta = \left[-2 \log\left(\frac{k/d}{3.7} + \frac{2.51}{Re\sqrt{\zeta}}\right)\right]^{-2} \qquad Re > 2300$$

$$Re = \frac{wd}{\nu}$$

The pressure drop, Δp per unit length is proportional to the friction factor, ζ, a measure of retardation of the flow by liquid viscosity and pipe wall roughness. The pressure drop, obviously, has no natural limits on the positive side and so does the friction factor.

The pressure drop depends on the density, ρ and the kinematic viscosity, ν of the liquid, the flow velocity, w, and the diameter, d and roughness factor k of the pipe.

For the sake of training a feed-forward neural network, examples were generated at random in the following range of inputs.

ρ	$0.5 - 1.5 \times 10^3$ kg/m^3
k	$0.1 - 1.0 \times 10^{-3}$ m
ν	$0.5 - 5.0 \times 10^{-6}$ m^2/sec
w	$1.0 - 10.0$ m/sec
d	$0.1 - 0.5$ m

This guarantees that $Re > 2300$.

3. The Levenberg-Marquardt method

The Levenberg-Marquardt method [3-6] was used to calculate the weights in the neural networks which minimised the sum of squares of errors. Most algorithms for least-squares optimisation problems use either steepest descent or Taylor-series models. The Levenberg-Marquardt method is a restricted step method, which uses an interpolation between the approaches based on the maximum neighbourhood (a "trust region") in which the truncated Taylor series gives an adequate representation of the non-linear model. The method has been found to be advantageous compared to other methods which use only one of the two approaches.

4. Results

The results give a clear indication of the applicability of the symmetric logarithmoid activation function.

4.1. Training the neural networks

A table of 200 training instances was created by random selection of inputs in the ranges stated in section 2. The pressure drop per unit length was tabulated in units of 10^3 Pa/m. Yet, it often had large values (greater than 10) which would require the argument of the symmetric logarithmoid to be very large. Therefore, the logarithm of the pressure drop per unit length (in units of 10^3 Pa/m) was tabulated instead. Similar results were obtained with actual pressure drop values.

With one hidden layer, the number of hidden nodes was varied between 2 and 7. The configurations with 5 or more hidden nodes resulted in good fits of the training data. The error square sums (SSQ) for (5,5,2), (5,6,2) and (5,7,2) were 0.189, 0.132 and 0.080 respectively. Networks with two hidden layers are not as easy to train as the ones with one hidden layer. This has been observed with the sigmoids, and was observed with this activation function also. The configuration (5,2,2,1) had a SSQ of 0.341, but (5,3,3,1) with 34 weights had a SSQ of 0.0583, less than 0.084, the SSQ of (5,5,1) with 36 weights. The SSQ for (5,4,4,1) was 0.0316.

4.2. Testing the trained neural networks

The trained neural networks were then tested with various velocities, while keeping other variables constant ($\rho = 10^3$ kg/m^3, $k = 10^{-4}$ m, $\nu = 10^{-6}$ m^2/sec, $d = 0.2$ m). A plot of the logarithm of pressure drop per unit length (10^3 Pa/m) vs velocity is shown in Fig. 1.

It can be seen that for various configurations shown in the legend of the figure, the predicted values are quite close to the analytical values. This accuracy is sufficient for engineering purposes. Typical error is about 0.05, which is about 2.5% of the range on the vertical axis.

5. Conclusions

The feed-forward networks trained with the symmetric logarithmoid activation function performed well in the testing phase. Convergence during training was faster than is usually encountered with sigmoids. The weights generated after training were never very large, although that happens often with sigmoids.

The symmetric logarithmoid, thus provides a feasible activation function for the neurons, instead of the sigmoid, when the outputs are not in a well-defined limited range.

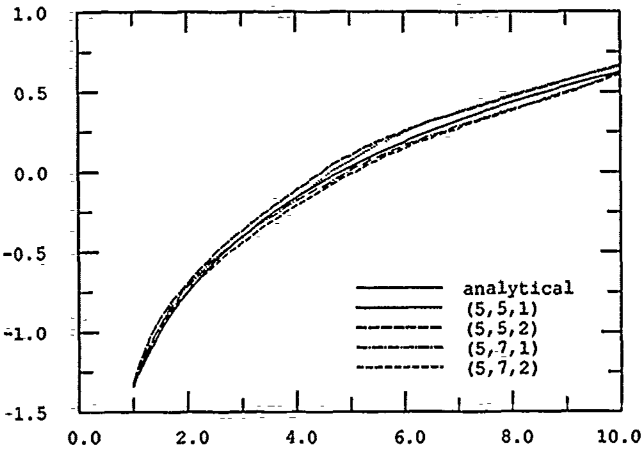


Figure 1. Logarithm of pressure drop per unit length vs velocity.

References

1. Bulsari, A. and H. Saxén,
"Applicability of an artificial neural network as a simulator for a chemical process",
Proceedings of the fifth International Symposium on Computer and Information Sciences, Nevşehir, Turkey, (October 1990) 143-151.

2. Streeter, V. L.,
"Fluid mechanics",
McGraw Hill Kogakusha, Tokyo (1962) 215.

3. Levenberg, K.,
"A method for the solution of certain nonlinear problems in least squares",
Quart. Appl. Math., 2 (1944) 164-168.

4. Marquardt, D. W.,
"An algorithm for least-squares estimation of nonlinear parameters",
J. Soc. Indust. Appl. Math., 11 (June 1963) 431-441.

5. Fletcher, R.,
"Practical methods of optimization, Vol. 1, Unconstrained optimization",
John Wiley and Sons, Chichester (1980) 82-88.

6. Gill, P. E., W. Murray and M. H. Wright,
"Practical optimization",
Academic Press, London (1981) 136-140.

Robot Vision Based on Coarsely-Grained Multi-Processor Systems

Volker Graefe

Institut für Meßtechnik
Universität der Bundeswehr München
8014 Neubiberg, Germany

Abstract

Concepts are introduced which allow robot vision systems to be designed according to the inherent structure of the task of vision. Practical results obtained with such systems are presented.

Introduction

Coarsely-grained multi-processor systems have been demonstrated in numerous practical applications to be particularly well suited for building powerful robot vision systems <Dickmanns, Graefe 88a, b>. The real-time vision system, BVV 1, is an early example. Conceived in the late seventies, it uses only a few 8-bit microprocessors (Intel 8085A) which by today's standards are rather weak and slow devices. Nevertheless, the BVV 1 as a system was demonstrated in 1982 to be sufficiently powerful for solving a demanding task in robot vision. the stabilization of an inverted pendulum <Haas 82, Graefe 83>.

Its successor, the BVV 2, was conceived in the early eighties <Graefe 84>. Based on much stronger 16-bit processors, Intel 8086, it enabled, for instance, the experimental vehicle VaMoRs in 1986 to follow a road at a speed of 96 km/h, making it by far not only the world's fastest fully autonomous road vehicle, but also exclusively the only one whose speed was limited by its engine and not by its vision system <Zapp 88>. The BVV 2 is an open system based on the standard Multibus I and using commercially available single board computers. This made it easy to replace its parallel processors by more powerful ones later. Such an improved version of the BVV 2 has been used for detecting and classifying obstacles on the road while approaching them with a speed of about 50 km/h on an unmarked road <Regensburger, Graefe 90; Solder, Graefe 90>.

Recently, the BVV 3 has become operational <Graefe 90>. It employs Intel microprocessors 80286 or 80386, augmented by a custom-designed coprocessor for feature extraction. The BVV 3 is intended to be used, for instance, in future experiments with an autonomous vehicle participating in ordinary highway traffic <Graefe, Kuhnert 88>. The BVV 3 should generally perform feature extraction about 5 to 10 times faster than the BVV 2, depending on the task. Tests have indicated that in some vision related applications it is actually more than 100 times faster than its predecessor.

It should be noted that the good performance of the BVV-systems is not a result of utilizing any particularly fast electronic components, but rather of a system architecture that matches the inherent structure of the task of robot vision.

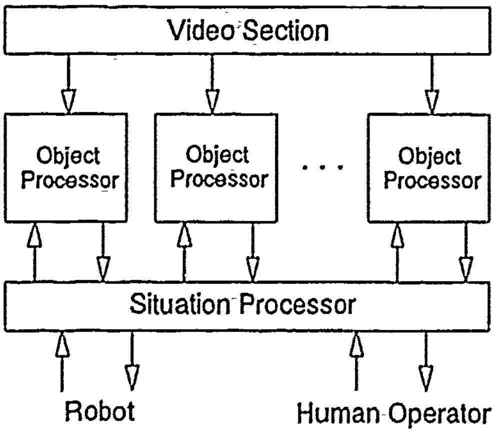


Figure 1
Conceptual structure of object oriented vision systems

System Architecture

A robot's environment contains a limited number of physical objects that are in some respect relevant for the operation of the robot. Among them could be landmarks, obstacles, the pathway, objects to be grasped, or various other objects. At any given moment only selected ones of these objects need to be monitored simultaneously, provided the robot is able to switch its attention from one object to another one within a fraction of a second.

A robot vision system should, therefore, have the ability to observe a small number of objects, say half a dozen, simultaneously. At the same time it should be able to maintain internal models of a slightly larger number of objects existing in the robot's environment but not demanding immediate attention according to the perceived situation.

If the structure of the task to be handled is reflected in the structure of the vision system a particularly high degree of efficiency may be expected. This has been discussed in greater detail in <Graefe 89>. In short, one "object processor" each should exist within the vision system for each relevant object in the robot's environment. Figure 1 shows the conceptual structure. It comprises a video section, a number of object processors, and a situation processor.

Video Section

The cameras, digitizers, and a means for distributing the video data to the object processors make up the video section. In addition, it may contain camera control devices like platforms or lens controllers. Each object processor has direct access to the digital video data. For the efficiency of the vision system it is of utmost importance that all object processors have independent access to these data. Therefore, each object processor should have its own image memory. A single shared image memory is, of course, cheaper, but it tends to create a severe bottleneck.

Object Processors

Each object processor receives digital image data from one or more cameras and outputs a description of one particular external object. The description may relate to the shape, location, state of motion, or other characteristics of the object. An object processor is a conceptual entity, not a physical one. In principle, it does not matter if such an object processor is implemented on exactly one piece of hardware, or if it contains several computing elements, or if several object processors share one computing element. Given the present state of microprocessor technology, an object processor for a typical robot vision application may be implemented on one to three microprocessors.

Situation Processor

The object descriptions are fed into a situation processor. This is, again, a conceptual entity. It will typically be implemented on a number of microprocessors. The main task of the situation processor is to deal with the interactions of the external objects with the robot and among each other. Besides, it assigns objects to the object processors. The situation processor also interfaces with the human operator, exchanges sensor and control data with the robot, and controls the video section, if necessary. The complexity of the situation processor's task depends highly on the complexity of the robot's task and on the environment the robot is supposed to operate in. Following an empty road, for example, involves hardly a situation to speak of, because the road is the only external object. On the other hand, driving autonomously in ordinary road traffic will certainly require a very powerful situation processor.

Implementation

The vision system BVV 2 has been designed to support structures as shown in Figure 1. Its video section contains four independent video channels allowing the signals from up to four cameras to be processed simultaneously. Each object processor typically comprises two parallel processors, one for a 2-D object model and the associated feature extraction, and one for a spatio temporal object model. The situation processor may be implemented partly on a PC AT and partly on a parallel processor within the BVV 2.

An implementation on the BVV 3 could be quite similar. But the full potential of the BVV 3 may be realized if all its parallel processors are used for feature extraction and 2 D object models. The spatio temporal models and the situation processor may then be implemented on separate hardware, possibly a transputer network.

Two new implementations of the object oriented architecture are currently being developed. Both of them will be based on a standard PC equipped with a commercially available video digitizer and a few custom built microcomputers to be utilized for feature extraction and 2-D object models. In one implementation the Intel 376 microprocessor will be used with an expected performance similar to the BVV 2 level and its software largely compatible with the BVV 2. The other new implementation will be based on the Intel 960. Its performance is hard to predict, hopefully it will approach the level of the BVV 3 at a lower cost and smaller size.

References

Dickmanns, E.D.; Graefe, V. (1988a): Dynamic Monocular Machine Vision. Machine Vision and Applications 1 (1988), pp 223-240.

Dickmanns, E.D.; Graefe, V. (1988b): Applications of Dynamic Monocular Machine Vision. Machine Vision and Applications 1 (1988), pp 241-261.

Graefe, V. (1983): A Pre-Processor for the Real-Time Interpretation of Dynamic Scenes. In T. S. Huang (ed.). Image Sequence Processing and Dynamic Scene Analysis, Springer-Verlag, pp 519-531.

Graefe, V. (1984): Two Multi-Processor Systems for Low-Level Real-Time Vision. In J. M. Brady, L. A. Gerhardt and H.F. Davidson (eds.). Robotics and Artificial Intelligence, Springer-Verlag, pp 301-308.

Graefe, V. (1989): Dynamic Vision Systems for Autonomous Mobile Robots. Proceedings, IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89). Tsukuba, pp 12-23.

Graefe, V. (1990): The BVV-Family of Robot Vision Systems. In O. Kaynak (ed.). Proceedings, IEEE Workshop on Intelligent Motion Control. Istanbul, pp IP55-IP65.

Graefe, V.; Kuhnert, K.-D. (1988): Towards a Vision Based Robot with a Driver's License. Proceedings, IEEE International Workshop on Intelligent Robots and Systems, IROS '88. Tokyo, pp 627-632.

Haas, G. (1982): Meßwertgewinnung durch Echtzeitauswertung von Bildfolgen. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.

Regensburger, U.; Graefe, V. (1990): Object Classification for Obstacle Avoidance. SPIE Symposium on Advances in Intelligent Systems. Boston, November 1990.

Solder, U.; Graefe, V. (1990): Object Detection in Real Time. Proceedings of the SPIE Symposium on Advances in Intelligent Systems. Boston, November 1990.

Zapp, A. (1988): Automatische Straßenfahrzeugführung durch Rechnersehen. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.

Acknowledgements

Part of the work reported has been performed with support from the Ministry of Research and Technology (BMFT) and from the German automobile industry within the EUREKA project PROMETHEUS and other projects.

Markov Random Field Models and Parallel Algorithms for 2D Motion Analysis

Fabrice HEITZ, Etienne MEMIN, Patrick BOUTHEMY
IRISA/INRIA, Campus Universitaire de Beaulieu
35042 Rennes Cedex, France

Abstract

The use of Markov Random Field (MRF) statistical models has recently brought new powerful solutions to classical image analysis problems. In recent papers, we have presented a new class of spatio-temporal MRF models which have been applied with success to different basic tasks in visual motion analysis. In this paper parallelization methods for those relaxation algorithms are investigated and a new hierarchical approach, based on the interactions between different relaxation processes running in parallel at different scales is presented. The hierarchical method has been simulated in the case of optical flow estimation. It shows good performances (quality of the estimates, gain in number of iterations) when compared to sequential algorithms.

1 Introduction

Markov Random Fields (MRF) models have been successfully introduced in several important issues of still image analysis, such as image restoration, segmentation or edge detection, [6] Our group has recently extended the MRF models to the analysis of image sequences for motion detection, [3], optical flow estimation, [8] and motion-based segmentation, [5]. In visual motion analysis, MRF appear as an efficient and powerful tool for combining spatial and temporal information. For details concerning the models designed in each case, we refer to [3, 8, 5].

The subject of the paper is the study of parallel approaches for the relaxation algorithms, [6] associated to MRF in image analysis Indeed, one very attractive property of MRF is that, though the models are global and non linear, the involved computations remain local and are intrinsically parallel. We present here an approach based on relaxation algorithms running in parallel at different scales and interacting periodically. This parallel algorithm is compared to sequential stochastic or deterministic algorithms in the case of optical flow computation. It exhibits fast convergence properties and only requires a small number of processors.

2 Markov Random Fields for motion analysis

The modeling and analysis of images by MRF's has been discussed extensively in the litterature, [6]. To extract labels describing motion from image sequences, observations related to the spatio-temporal variations in the image sequence are combined with a priori generic knowledge on the expected solution, in order to derive estimates of the unknown labels. The labels are binary features in motion detection, vectors in optical flow estimation and region numbers in motion-based segmentation, [3, 8, 5]. MRF models describe the *local* statistical interactions between these different variables. When a maximum a posteriori (MAP) estimate of the unknown label variables is looked for, MRF-based image analysis reduces to the minimization of a global energy function U which depends on the whole observation and label field, [6]. Minimizing the global energy function U is an intricate problem : the number of possible label configurations is generally *very large* and the global energy function U may exhibit numerous local minima. Computationally demanding stochastic relaxation algorithms are therefore generally necessary to compute exact MAP solutions. Deterministic descent algorithms such as ICM, [2] can often be used instead, when a good initial guess is available.

A simple model for optical flow measurement

We consider here a very simple model for optical flow computation, which will be used for comparison purpose between the parallel and sequential versions of stochastic relaxation (for more sophisticated models including discontinuities and occlusion processing see [8]).
Let $f(s,t)$ denote the observed intensity function, where $s = (x,y)$, $s \in S$ designate the 2D spatial image coordinates and t the time axis. The velocity vector at point s is denoted $\vec{\omega}_s(u_s, v_s)$, $u_s = \frac{dx}{dt}(s), v_s = \frac{dy}{dt}(s)$ and $\vec{\omega} = \{\vec{\omega}_s,\ s \in S\}$. In the model considered here velocities are defined on the same grid S as the pixels and the velocities are discretized according to a discrete state space $\mathcal{W} = (-u_{max} : u_{max}, -v_{max} : v_{max})$ with a step size of δ. The MRF model is associated to a 8-neighbourhood and specified by following energy function :

$$U(f,\vec{\omega}) = \sum_{s \in S} \{ f(s,t) - f(s + \vec{\omega}_s.dt, t + dt) \}^2 + \alpha^2 \sum_{(s,t)\ neighbours} \| \vec{\omega}_s - \vec{\omega}_t \|^2 \qquad (1)$$

The first term in the energy (known as the "displaced frame difference") expresses the constant brightness assumption for a physical point over time. The second term balances the first one through weighting parameter α, it can be interpreted as a regularization term which favours smooth solutions. This discrete state space in the optical flow estimation problem leads to a complex energy landscape showing numerous local minima. Hence this model is as a good benchmark for parallel stochastic relaxation algorithms.

The stochastic relaxation algorithms are based on the generation of realizations of Markov Chains whose limit distribution correspond to the Gibbs distribution $p(f\ \vec{\omega}) = \frac{1}{Z}\exp -U(f,\vec{\omega})$, [6]. Two basic relaxation algoritms are generally used : the Metropolis algorithm, [1] or the Gibbs sampler, [6]. In the case of optical flow computation, the Gibbs Sampler can be described as follows, [6] :
let $(n_1, n_2, ...n_t)$, $n_t \in S$ be a sequence in which the sites of the vector field $\vec{\omega}$ are visited for updating (raster scan will be considered here). The corresponding label configurations are denoted $\vec{\omega}^{(1)}, \vec{\omega}^{(2)}, .., \vec{\omega}^{(t)}$. $\vec{\omega}^{(0)}$ is an initial configuration chosen at random. Let $T(t)$ be a decreasing sequence of temperatures. At time t site n_t of $\vec{\omega}^{(t-1)}$ is updated by drawing a sample from the local characteristics of the Gibbs distribution : $p_T(f,\vec{\omega}) = \frac{1}{Z(T)}\exp\frac{-U(f,\vec{\omega})}{T}$. It is straightforward to show that this computation is local, thanks to the markovian property of the model and only involves site n_t and its neighbours. $\{ \vec{\omega}(0), \vec{\omega}(1), ...\vec{\omega}(t), ... \}$ defines a Markov Chain whose convergence properties have been studied extensively, [1, 6]. A logarithmic decreasing temperature schedule is required to ensure convergence to a global minimum of the energy function. To save computation time, we have considered less conservative exponential schedules of the form $T(t) = T_0.A^t$, $A < 1$ (which are often used in practice).

3 Parallel algorithms for stochastic relaxation

Parallel implementations of stochastic relaxation have been considered for global optimization in applications such as computer-aided circuit design, [1] or image processing, [6, 9, 4] Until now, three main approaches have been investigated . *parallelized Markov chains, [1, 7], simultaneous updating of the image sites, [6, 9]*, and *parallelization of the local label updating. [4]*

Parallelized Markov chains have been proposed by Aarts *et al* for global optimization problems based on simulated annealing and the Metropolis algorithm, [1]. The basic principle of the approach is to run in parallel several relaxation algorithms, each of them exploring differently the space of all possible label configurations and interacting from time to time. This class of algorithm may be run on a MIMD or SIMD machine by assigning each available processor to a different relaxation algoritm. The interaction are based on periodic transfer of global label configurations between the different processors A similar method has recently been studied by Graffigne, [7]

Our own approach of the problem is also based on parallelized Markov Chains but derives profit from a hierarchical representation of data and labels, which leads to an efficient parallel annealing algorithm with a speed-up larger than the number of processors. The method can be outlined as follows . a data pyramid is first constructed by low-pass filtering and subsampling the images of the sequence. Accordingly, a hierarchical representation is also considered for the label fields which are estimated in parallel at different resolution levels on reduced grids (Fig. 1). Relaxations are performed in parallel by assigning one processor at each level of the pyramid.

Following [7], the different relaxations are performed at fixed temperature. To the low resolutions levels of the hierarchy are associated high temperatures. At low resolutions, coarse estimates of the label field are visited. Since the total number of possible label configuration is reduced at those scales, the coarse configuration space can be visited very efficiently. A high temperature enables to escape from local minima of the energy function. To the intermediate resolution levels are associated lower temperatures. At these levels, the relaxation process becomes more sensitive to local minima and visits the large or medium scale valleys of the energy landscape, [7]. At the finest resolution level a temperature close to 0 is adopted. Very low temperature in the stochastic relaxation corresponds to nearly deterministic descent of the energy function : the estimation is ultimately refined at that level.

The explorations at the different levels are cooperative . every p iterations - one iteration corresponding to a full sweep on the image - a processor attempts to transfer a small label block to the next finer level (Fig. 1). The interaction process is controlled as follows . a block B_k^r at resolution level r is interpolated at level $r-1$ using simple repetition of the label estimates at the missing positions of level $r-1$ (Fig. 1). The energy of the resulting local configuration is computed and compared to the energy of the corresponding block B_k^{r-1} at the finer resolution $r-1$. The local label configuration on the block B_k^{r-1} is replaced by the interpolated configuration of block B_k^r if the latter is better than the former, that is, if its energy is lower. Local energy on blocks are obtained at reduced additional cost from the local characteristics of the Gibbs distribution.

Experimental results

Experiments have been carried out, in the case of the discrete optical flow measurement model, on several synthetic and real world sequences. Three algorithms have been simulated : a sequential stochastic relaxation (*SSR*) based on the Gibbs sampler with an exponential schedule ($A = 0.97, T_0 = 300$), a sequential deterministic relaxation (*SDR*) algorithm known as ICM, [2], and the proposed parallel hierarchical relaxation (*PHR*). In our experiments the same parameter value α (equ. 1) was used for the potential functions at each level of the label pyramid ($\alpha^2 - 20$). In every case the convergence criterion was the same . the relaxation was stopped as soon as the maximum number of modified labels between two successive image sweeps (at the finest resolution level) went below some specified threshold (typically 10).

Figure 2 shows the energy plots obtained for a divergent motion on 64x64 images by the different algorithms. Four resolution levels are used in the pyramid and $u_{max} = v_{max} = 8$ in this case. *PHR* produces estimates close in quality to the one obtained by *SSR* (the energies of the final configuration are respectively 96700 and 87300). The slight degradation of *PHR* with respect to *SSR* originates from the block effect due to the interscale interaction mode. *PHR* exhibits fast convergence properties (similar to *SDR*, but with estimates of significantly better quality). On an average, the computational gain (in number of iterations) of *PHR* over *SSR*, over several sequences, was about 10. An implementation of the proposed algorithm on an iPSC/2 supercomputer is currently investigated.

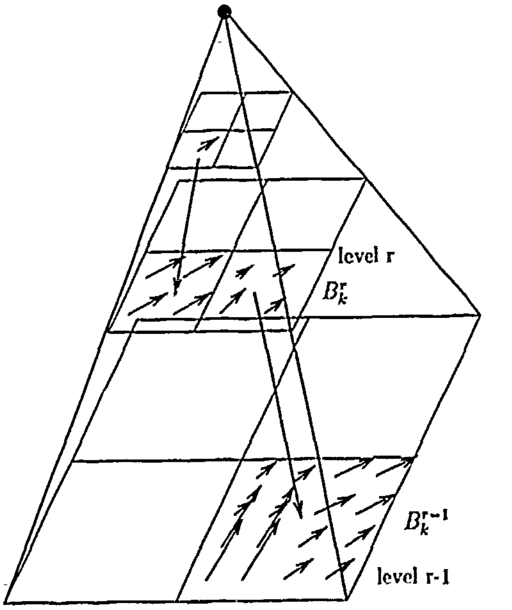


Figure 1 : Parallel Hierarchical Relaxation . the label pyramid

REFERENCES

[1] E.H.L. AARTS and P.J.M. van-LAARHOVEN, *Simulated annealing. theory and applications*, D. Reidel Publishing Company,1987.

[2] J. BESAG, On the statistical analysis of dirty pictures, *J. Royal Statist. Soc.*, Vol. 18, Serie B, No 3, 1986, pp. 259-302.

[3] P. BOUTHEMY and P. LALANDE, Detection and tracking of moving objects based on a statistical regularization method in space and time, in *Proc. First European Conference on Computer Vision*, Antibes, France, April 1990, pp. 307-311.

[4] H. DERIN and C.S. WON, A parallel image segmentation algorithm using relaxation with varying neighbourhoods and its mapping to array processor, *Comput. Vision, Graphics, Image Processing*, Vol. 40, 1987, pp. 54-78.

[5] E. FRANCOIS and P. BOUTHEMY, Multiframe-based identification of mobile components of a scene with a moving camera, in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, Hawaii, June 3-6, 1991.

[6] S. GEMAN and D GEMAN, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.6, No.6, Nov. 1984, pp. 721-741.

[7] C GRAFFIGNE, A Parallel Simulated Annealing Algorithm, *Technical Report*, CNRS, Université Paris-Sud, 1990.

[8] F HEITZ and P. BOUTHEMY, Multimodal Motion estimation and Segmentation using Markov Random Fields, 10th Int. Conf. Pattern Recognition, Atlantic City, Vol. 1, June 1990, pp. 378-383.

[9] D.W. MURRAY, A. KASHKO and H. BUXTON, A parallel approach to the picture restoration algorithm of Geman and Geman on an SIMD machine, *Image and Vision Computing*, Vol. 4, No. 3, 1986, pp. 133-142.

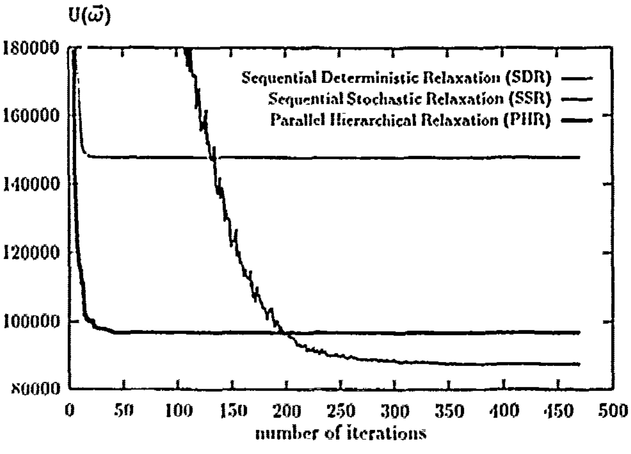


Figure 2 . Energy U($\bar{\omega}$) versus iteration number

PARALLEL COMPUTER ARCHITECTURES USING HETEROGENEOUS PARALLEL PROCESSING STRUCTURES FOR REAL-TIME VISION BASED ON-LINE INSPECTION OF MANUFACTURED PARTS [1]

E. HIRSCH, Ph. PAILLOU
Laboratoire des Sciences de l'Image et de la Télédétection, LSIT, Ecole Nationale Supérieure de Physique de Strasbourg, ENSPS, Université Louis Pasteur, 7, rue de l'Université, F-67000 Strasbourg, France.

C. MÜLLER
Institut für Algorithmen und Kognitive Systeme, IAKS, Fakultät für Informatik, Universität Karlsruhe (TH), Fasanengarten, D-7500 Karlsruhe 1, FRG.

U. LÜBBERT, V. GENGENBACH
Fraunhofer Institut für Informations- und Datenverarbeitung, IITB, Fraunhofer Str. 1, D-7500 Karlsruhe 1, FRG

Abstract.

In this paper, we present concepts related to the design of parallel computer architectures for use in real time vision applications and based on combinations of heterogeneous processing structures. The corresponding realizations will be described and the use of such systems illustrated by a representative application, e.g the on-line inspection of manufactured parts.

1 Introduction

Most of the current industrial applications of machine vision, relying on image processing systems with high computation power, are concerned with inspection, quality control, assembly, control of manufacturing processes, autonomous vehicle guidance and robotics. Optical sensing of the application environment, evaluation of the images of the scene and physical reaction after interpretation of the image contents are among the most effective and efficient means for the analysis of the environment and for acting in an appropriate way. Machine vision is thus a way to automate the applications previously described in a flexible and intelligent manner. The goal of computer or machine vision is to extract high-level information about the environment from the low-level information contained in one or a sequence of images of that environment. This should contribute to the design/development of so-called intelligent machines. Because of the amount of information contained in an image, machine vision calls for high computing power. Provided the required hardware is available, this leads to the development of vision systems with the aim to solve these problems automatically, with a computation speed compatible with the application to be solved [1], [2].

This naturally leads to the use of parallel computer architectures, and more precisely to heterogeneous pipelined structures. However the different systems, proposed for the industrial applications quoted above, are often not able to provide the performance and processing speed needed at reasonable costs. This is also the case for more sophisticated applications such as, for example, the automated vision based on-line inspection of manufactured parts in order to detect and locate different defects on the part under control for the optimization of production rate. Specific requirements for the vision systems running such vision tasks have thus firstly to be defined and secondly to be satisfied.

With the technology available today, in order to run the application in "real time", it is necessary to use heavily parallelism for the image processing and some higher level treatments required by the application. More specifically, one has also to match hardware to the envisioned processing. This leads to the design and implementation of heterogeneous pipelined structures, in which each component is optimized with respect to the type of processing it has to perform. Furthermore, the elements of the pipeline often imply an internal parallelism (for example, systolic processors, arrays of processors such as Transputer networks). Last but not least, the synchronized use of such systems calls for a carefully designed programming environment, relieving partly the user of the burden of programming such systems.

The following section introduces the definitions and concepts involved by the design and implementation of such machine vision systems. Section 3 describes the image processing systems already realized, using two different approaches. The first system, designed and built at the Fhg-IITB Institute, FRG, uses a modular functional approach, whereas the second one aims to implement a reconfigurable, flexible and fully programmable structure. Both systems are able to carry out most of the low level image processings in video real-time.

Development of methods and design of new architectures call thus for carefull studies and possibly experimentation. This should be facilitated by the set-up of development centers grouping hardware and software resources. A possible configuration of such an integrated development system for computer vision applications will be indicated.

The paper is concluded by an illustration of the use of such systems, an automated system as a solution to the 100 % control of manufactured parts in a FMS environment. The technique used is based on comparison between images acquired through a vision system and the corresponding data gained from a CAD system. Comparison takes places as well at feature level than at image level. All kind of inspections, ranging from conformity checking up to metrology, can be achieved through use of an user friendly planning system.

2 Machine Vision-Systems

In vision applications, two different processing steps can usually be defined.The first step deals with the so-called low-level image processing, which transforms an input image into a modified output image in order to enhance the quality of the input image (e.g. noise reduction and distortion correction) and prepares a subsequent feature extraction. Then some segmentation of the enhanced image is done in order to delimit different regions in the image. The extracted features represent (or model) some characteristic properties of the information content of the scene (perimeter, connexity, moments...). Thus, after having computed a description of the image content, the second step exploits this description in order to compute statements about the meaning of the image content. These statements are then further used for firstly defining and secondly carrying out the actions to be performed on the environment implied by the application. Thus, this interpretation of the image is used to achieve a well defined goal (e.g. recognizing objects, taking decisions, vehicle guidance, inspection, acting on the real-world,...).

Imaging devices (generally a CCD camera) produce as input for the machine vision system a pixel image. The purpose of the vision system is to transform the low-level information content of the input image into a number (as small as convenient) of high-level information pieces, in order to deliver a description of the real-world scene. This description can be further used to take a decision, which, in turn, leads to one or more actions to be carried out on the real-world through a feedback-loop. Instead of a feedback free processing model, a more complex processing scheme, including a feedback loop called interpretation loop, can be imagined.

Figure 1 above shows the model first proposed by Kanade [3], [4] and modified by Nagel [5], [6], [7].It is important to see that the model of figure 2 allows iterative image interpretation and eventual control of the data acquisition parameters. Furthermore, the model implies different types of knowledge for the interpretation process in both the image (low level) and scene (high level) domains. More and more industrial vision applications rely on such a processing scheme, as the examples in section 4 will illustrate.

[1] Part of the work described in this paper has been carried out under ESPRIT PCA Application A4137 and ESPRIT Project 2091 VIMP (" Vision Based On-line Inspection of Manufactured Parts. Comparison of CCD and CAD Images ").

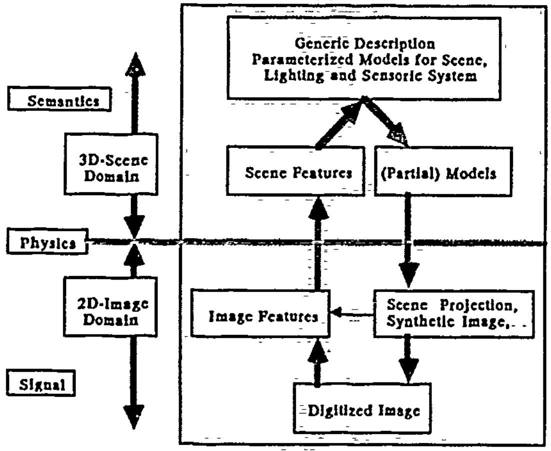


Figure 1 : Computer vision as an iterative interpretation process (after [6],[7]).

Image processing thus implies very different processing tasks with respect to applications. These image-processing tasks are very different in nature and imply very different data structures for the computations. From the system designer point of view, these levels are usually refered as low, medium and high-level image processing (see fig. 2). This sub-dividing of a complete vision task into three processing levels, reflects the nature of the operations to be carried out and the nature of the data structures used. Figure 2 defines the three processing levels and indicates how the different stages are combined in a sequential processing scheme, implementing the direct processing path of the model of figure 1 :

- Low level image processing transforms images into images. They are essentially pixel oriented operations . threshold, lowpass filtering, mathematical morphology.

- Medium level image processing needs images as input and gives features as output. Extraction of lists of contours is a widely known example.

- High level operations transform features into features. Other sources of information than the initial images can be used (a priori knowledge, databases, artificial intelligence). High-level processing also has in charge the control of the interfaces between the machine vision system and its environment (e g man-machine interfaces, communication network interfaces, actuator interfaces).

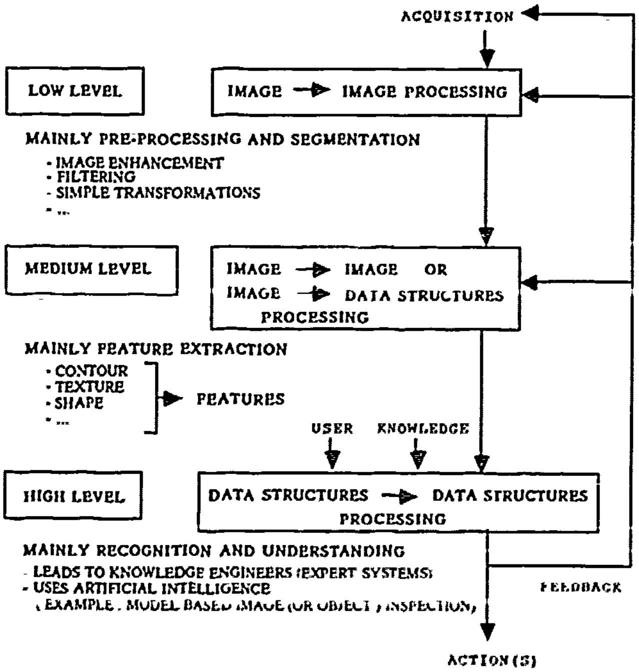

ACQUISITION
LOW LEVEL
IMAGE → IMAGE PROCESSING
MAINLY PRE-PROCESSING AND SEGMENTATION
- IMAGE ENHANCEMENT
- FILTERING
- SIMPLE TRANSFORMATIONS
- ...
MEDIUM LEVEL
IMAGE → IMAGE OR
IMAGE → DATA STRUCTURES
PROCESSING
MAINLY FEATURE EXTRACTION
- CONTOUR
- TEXTURE
- SHAPE
- ...
FEATURES
USER
KNOWLEDGE
HIGH LEVEL
DATA STRUCTURES → DATA STRUCTURES
PROCESSING
MAINLY RECOGNITION AND UNDERSTANDING
- LEADS TO KNOWLEDGE ENGINEERS (EXPERT SYSTEMS)
- USES ARTIFICIAL INTELLIGENCE
(EXAMPLE . MODEL BASED IMAGE (OR OBJECT) INSPECTION)
FEEDBACK
ACTION(S)

Figure 2 : Processing levels in machine vision.

Each processing level has to be implemented on hardware specifically tailored for it, in order to perform efficiently its tasks.Taking this matching into account leads then directly to the design of heterogeneous pipelined image processing structures for complex vision applications. The amount of data to be computed is decreasing from the low-level to the high-level stage, and less and less computing power is needed, but more ana more flexibility is required.

According to figure 2, the basic structure of an image processing system can also be defined around three parts, having each different requirements to the processing architecture :

- The acquisition, pre-processing and restitution of images.

- The processing in the image domain or the so-called iconic processing step. In this case, the needed algorithms and data structures are nearly directly related to the pixel organisation in the image.

- The model based exploitation of image primitives and content or the so-called symbolic processing step. In this case, the data structures implied are almost exclusively a function of the type of computation to be carried out.

Three basic structures (for a discussion, see for example, [2], [9]), for so called general purpose sytems, are today clearly emerging .

Arrays of mesh-connected processors with more or less computation power,

- pipe-lines and cascaded structures, chaining different hardware modules with more or less flexibility,

- bus-oriented processing structures with powerfull, more or less specialized micro--processors.

For special purpose processing, systolic arrays or dedicated LSI or VLSI integrated circuits can also be very efficiently used [10]. However, bootlenecks are usually encountered when the low-level pixel data has to be matched with the high-level data structures [11], [12]. It seems therefore not possible to design a system able to perform well for all the algorithms needed in computer vision. However, heterogeneous structures for the three levels of processing, with an integrated control strategy and incorporating different modes of parallelism, appear to be today the best compromise. With respect to the use of such systems, two approaches, exemplified in the next sections, are possible:

- The first is based on a modular functional approach, where the user has just to choose the appropriate chaining and parameters for image processing operations,

- The second is based on a fully programmable approach, where the user can develop his own software for all the system components and organize his own data flows and image processing sequences.

3 Parallel Architectures for Computer Vision

Sophisticated vision tasks like those mentioned in the introduction imply numerous requirements for a vision system, which have been implemented into the systems to be described :

- Bottlenecks should be avoided either by extending or by specially organizing the system. Addition of supplementary modules for parallelization and the support of general purpose processors with special hardware processors help to remove these bottlenecks

- Image acquisition should be separated from the image processing and interpretation steps (see figures 1 and 2) in order to process data at maximum speed.

- Organization of data should not be fixed in order to give the system the greatest flexibility.

- With respect to signal acquisition, storage and display, different sensors must be available, differing in format but also in physical nature, in order to be able to process multidimensional images and multisensorial signals.

- Processing of images, isolated, endless or in sequence, should be possible. Furthermore, the system hardware must be able to adapt to particular classes of tasks, varying in complexity, in order to allow a cost effective match of the system to the application to be solved. Bus-oriented multiprocessor architectures whose modules are the processing elements can efficiently be used. The modules can be classified into three categories :

. Video I/O modules for acquisition, storage and display,

. pixel oriented image processing modules working synchronously with the image acquisition rate and offering the possibility of parallel and pipelined processing, for data compression in the iconic stage (image processing and feature extraction),

. not pixel oriented data processing modules working asynchronously with the image acquisition rate, for statement generation in the symbolic stage.

- The system should offer the possibility of being used as a development system (e.g. for testing the image processing or evaluating the processing steps needed for solving a specific vision task), or as part of a workstation (e.g. for the solution of real problems in industrial environment). A hierarchical layered approach can be retained for the software organization, which does not require the user to be a low-level programming expert. Furthermore, an exhaustive package of image processing and evaluation programs should be provided. The development of problem oriented software should be possible using high level languages and ergonomic man-machine interfaces helping the user in his application software developments.

The architecture of the parallel vision machines developped (see fig. 3 for an example) implements the requirements stated in the preceding section and is matched with the three processing levels described in fig.2. The image to be interpreted is sent to the low-level processing unit. The result is then fed to the medium-level processing unit which produces features for the high-level processing unit.

A feedback-loop has to be foreseen from the high-level stage to the medium and low-level stages (use of high-level knowledge to guide the processing of the lower stages). The general architecture looks like a system of pipelined processing elements, each of the processing units being able to make use of its own internal parallelism, adapted to the type of computation to be performed (leading to the concept of multi-parallelism).

The Low-Level Processing Units. The low-level processing is performed either with a VISTA system (Visual Interpretation System for Technical Applications) developped at the Fraunhofer Institut IITB in Karlsruhe, FRG [13], [16] -illustrating the modular functional design approach- or with dedicated hardware build around a mesh of 1 bit processors at ENSPS in Strasbourg, F [14], [15], [16] -exemplifying the reconfigurable, flexible and programmable design approach.

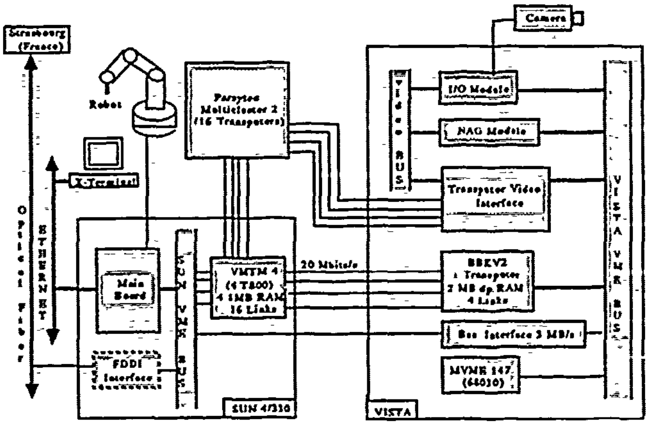


Figure 3 : The machine vision system developped at IAKS

The Medium-Level Processing Unit A network of sixteen T800 Transputers (Multicluster 2 from Parsytec or T-Node from Telmat) has been chosen as a medium-level processing unit The Transputer technology has been retained because of its flexibility allowing to efficiently accomodate the computation needs, as the complexity of processing increases from the low-level to the medium-level At this level, OCCAM has been used as the programming language for Transputers, because it provides the advantages of a high-level language and permits to easily exploit concurrency The OCCAM compiler is also very well optimized as regards executable code size and execution time. As the inputs of the medium-level stage are images, an interface between the low-level stages video-bus and Transputer links of the network has been developped to optimize data transfer rates between the low and medium processing stages. The integration of the Transputer based systems is shown on fig. 3. The integration of the specialized I/O interfaces, the low level processor modules, the medium level Transputer networks and the cooperation between the two levels is achieved through partitioning of the processing algorithms over the two stages, taking into account the characteristics of each processor and the kind of algorithms that each processor can optimally execute. Consequently, the feasibility of an efficient combination of the SIMD and MIMD approaches for low and medium-level image processing has been confirmed. The processing capabilities in actual applications has also been verified (see section 4.). This should lead to a tentative definition of the domains of use of each type of parallelism, leading to rough rules for finding an optimal architecture for a given application and/or a given time performance requirement.

The High-Level Processing Unit. The high-level processing unit is a SUN 4/330 workstation. This workstation is receiving features from the medium-level processing unit in order to interpret the initial real-world scene content. Programs running on this workstation are written in C++, an object oriented programming language, which allows a flexible and easy manipulation of the feature data coming from the medium-level stages and usually packed into specific objects. The SUN workstation initializes, configures and loads the two vision systems and the Transputer networks with appropriate algorithms and is used as an interface between the machine vision system and its environment (Ethernet, robot control,) Communication (and feedback loop) with the low/medium level stages are done using either specially developped Transputer based interfaces, or a Transputer board (BBKV2 from Parsytec) with dual-ported RAM between the Transputer memory bus and the VME or Multibus-bus of the low level stages, or a bus interface between the workstation bus and the low-level system buses. Communications with the Transputer network are performed with the help of another Transputer based board (VMTM from Parsystec).

When planning a vision system, it is necessary to provide "universal" tools in order to relieve partly the user of the burden of programming such complex systems. Due to the envisioned applications, a great variety of tasks results, leading to very distinct operating facilities and system performance. Furthermore, in laboratory, tools are needed for the development of methods and for the implementation of algorithms in order to test their time and logical behaviors. The tools must provide facilities enabling the user to take advantage of the underlying hardware without special knowledge about this hardware. On the other hand, the requirements for industrial applications are high performance, short reaction time and minimum user interaction. In our systems, image processing steps can be run autonomously or interactively. The compiled code for the processing steps is downloaded into the user program, where it is parameterized and started. Once a step has been started, it continues autonomously, independently of the calling user program and synchronized with it via an autonomous control software. The software structure is also in charge of the high-level unit. It can be seen as the overall system management software, including the man-machine interface. This man-machine interface is running on the SUN workstation and is developed under X-windows and OSF/MOTIF, which allow the development of menu-driven applications.

A Distributed (Transputer Network Based) Development Tool. Within the Parallel Computing Action Application PCA 4137, two Transputer networks will be interconnected using an optical fiber. The two systems, located the first in Strasbourg -France, and the second in Karlsruhe -Germany (see fig. 3), will be interconnected through an already existing optical fiber between the two institutes involved in the application. The protocol chosen for data exchange is the newly available FDDI standard. The performances of this standard are such that the overall performance of the distributed transputer network should not be lowered. The connection also punctually allows the access by one institute to the whole shared resources, when high computing power is required. The distributed system can then be seen as a very complete development system for vision applications, joining the resources of the two institutes and enabling the use by each institute of the two different low-level processing units.

4 Applications

The machine vision systems described are able to deal with quite a lot of applications. A brief description of one of these follows. An example of more sophisticated application, currently under development and to be run on the described hardwares, is the inspection using computer vision, on-line in the manufacturing environment, of parts being manufactured [17],[18]. Inspection takes place through comparison of CCD images and corresponding conceptual representations gained using the CAD model of the piece. Comparison takes places as well at feature level as at image level. All kind of inspections, ranging from conformity checking up to metrology, can be achieved through use of an user friendly planning system Due to the limited resolution of the imaging sensors used today for digital image processing, the sensor is moved in order to scan larger workpieces in their entire extent. CAD-based knowledge is also required and used for efficient performance of such hybrid mechanical and electronical inspection tasks. To inspect the geometrical properties, the image of the current field of view is compared with the information stored in the data-base of the associated CAD system, with or without use of structured light. Comparison takes place after segmentation and registration of the actual image with a synthetic projection of the CAD module. Furthermore, the 3D data coming from the CAD system are used to generate a 2D representation corresponding to the angle of view of the sensor. The output of the inspection stage is used for retrofitting by the manipulator in case of a possible remanufacturing.

The advanced inspection system described consists of knowledge processing and engineering at high level, but the first layers will consist of simulation, preprocessing, feature extraction (e.g. filtering, identification, segmentation) followed by recognition (registration, inspection) and decision taking. Again, the inspection task is the chaining of different processing tasks. The three processing levels defined in section 2 and implemented in the systems described sections 3 can easily be recognized. It is noteworthy to see that the same model can be used for a wide range of applications (pick and place, autonomous vehicle guidance, robotics, for example) just by changing the a-priori knowledge (e.g. the CAD database).

The comparison itself may take the form of a simple distance measure (for example as is suitable for difference determination between the original and real parts). Alternatively, a more complex processing such as feature descriptor based methods (modified Fourier descriptors, for example) may be used to compare the images obtained from the CCD sensor and those synthesized from the CAD data-base.

In order to facilitate the implementation of such a vision application, it is definitively necessary to integrate, as transparently as possible for the user, the vision system in the complete inspection system, as well with respect to the hardware components as with respect to the overall manufacturing environment.

Up to now, the design of vision systems was dominated by the image processing and interpretation methods themselves. In the future, however, it will be necessary that not isolated vision systems but complete systems, where vision is only a component, are optimized, as, for example, in the field of Computer Integrated Manufacturing (CIM). The resulting increased requirements for the number and performance of the used sensors will further lead to new evaluation criteria, which must be taken into account for the design of vision systems. The systems should also be integrated in local array networks (ETHERNET or the newer FDDI) in order to exchange data with other sub-systems. Integration within an overall manufacturing environment should also be foreseen, for example by using MAP (Manufacturing Automation Protocol). Development of methods and design of new architectures call thus for carefull studies and possibly experimentation. This should be facilitated by the set-up of development centers grouping hardware and software resources. A possible configuration of such an integrated development system for computer vision applications has been indicated in simplified block-diagram form in fig 3.

5 Conclusion

A model of machine vision architecture has been implemented using different functional levels, leading to a general purpose parallel computer for machine vision. The machines are based on pipelined low, medium and high level processing units, each having its own well-suited parallelism.

As a result of experimentation carried out with the vision described systems, the assessment of SIMD and MIMD parallel architectures for low level processing tasks and the development of knowledge based methodologies for high level applications, should allow improved performance and competitiveness in the development of a wide range of image understanding applications, all based on two common, very flexible parts:

- A kernel of image processing routines for software. A general software kernel for vision systems would allow the designer to go from one application to the other only by changing the knowledge specific to the application (model of the scene under examination and of its content), instead of changing the whole software.

- An heterogeneous parallel machine for hardware. The demonstrated feasibility of programmable nardware for image processing enables faster implementation of a wide range of industrial applications. With only one programmable parallel machine, it will be possible to solve a number of disparate computer visions applications related to various domains.

However, further work is necessary in this field, as research must be continued with the aim in particular of building a better SIMD machine and of improving the coupling with the MIMD machines. In addition, it is necessary to develop an extensive software support, as software implementation of algorithms will definitively be the major cost factor in developing new applications. Furthermore, the future realization of vision systems will be influenced in particular by the integrability of vision systems as sub-systems in complex applications, possibly in the feedback path of the control loop (see figure 2) of these systems in order, in particular, to control the sensors and/or the lighting conditions (active sensing) for an automated, dynamically optimized, data acquisition.

Such vision systems are likely to run a lot of applications, among which comparison of CCD and CAD data for automated workpiece inspection was described. The type of computer vision architectures presented seems to be very promising for the future, as artificial vision is becoming a more and more important. The approach described could also lead to more "intelligent" robots, using such vision systems to understand their environment.

Acknowledgements

The authors are greatly indebted to Prof. Dr. H.-H Nagel for having provided them the insight in computer vision. They also wish to thank their colleagues at the ENSPS/LSIT and Fhg-IITB Institutes and for the kindful permission to use some of the illustrations presented in this paper.

References

[1] R.K. Miller, Machine Vision for Robotics and Automated Inspection, 3rd ed, Fort Lee, NJ . Technical Insights, 1985, 3 vols.

[2] J L C Sanz, Introduction to the Special Issues on Industrial Machine Vision and Computer Vision Technology, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, No. 1, January 1988, and No. 3, May 1988.

[3] T. Kanade, Proc. Int. Joint Conf. Pattern Recognition, Kyoto, Japan, November 7-10, 1978, p. 95-105.

[4] T. Kanade, Computer Graphics and Image Processing, 13, 1980, p. 279-297.

[5] H.-H. Nagel in "Angewandte Szenenanalyse", J.P. Foith (ed.), Informatik-Fachberichte 20, Berlin Heidelberg New York. Springer-Verlag, 1979, p. 3-21.

[6] H.-H. Nagel in Informatik-Fachbericht 112, W. Brauer and B. Radig (eds.), Berlin Heidelberg New York Tokyo. Springer-Verlag, 1985, p. 170-199.

[7] H. H. Nagel in "Fundamentals in Computer Understanding . Speech and Vision", J P Haton (ed.), Cambridge University Press, 1987, p.113-139

[8] H H Nagel, Proc 1st International Exhibition and Conference on Applied Vision Systems Vision '88, Stuttgart, FRG, 1988, p.15-23.

[9] J.L.C Sanz, Machine Vision and Applications, 2, 1989, p. 167-173.

[10] T.J. Fountain, Proc. 8th International Conference on Pattern Recognition ICPR-86, October 1986, Paris, p.24-33

[11] M.J.B. Duff, in "From Pixels to Features", J.C. Simon (Ed), Elsevier Science Publishers B.V. (North Holland), 1989, p. 403-413.

[12] M.J.B. Duff, Proc. 10th International Conference on Pattern Recognition ICPR-90, june 1990, Atlantic City, USA, p.24-33.

[13] D. Paul, W. Hättich, W. Nill, S. Tatari, G. Winkler, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.10, No. 3, May 1988, p.399-407.

[14] Perucca G., Giorcelli S., De Couasnon T., Hirsch E., Mangold H., in "Putting Technology to use", CEC/DGXIII Ed., North-Holland, 1988, p 543-561.

[15] Pierre F., Herve Y., Eugene F., Draman C., Wendling S., Proc. 2th PIXIM Conference, Paris, 1988.

[16] E. Hirsch, in "From Pixels to Features II, Parallelism in Image Processing", ESPRIT BRA3035 Workshop, Bonas, France, 1990, p. 329-348.

[17] E. Hirsch, U. Lubbert, in "Computer Integrated Manufacturing", L. Faria and W. Van Puymbroeck (Eds.), 1990, p. 76-90, Berlin Heidelberg New York . Springer-Verlag.

[18] E. Hirsch, " Vision Based On-line Inspection of Manufactured Parts " in " Computing with Parallel Architectures", Amsterdam, Kluwer Academic Publishers, to appear in 1991.

PARALLEL ALGORITHMS FOR LOW LEVEL VISION ON A CONNECTION MACHINE(CM2)

Josiane Zerubia and Florimond Ployette

INRIA-Sophia - 2004 route des lucioles - 06560 - Valbonne.
FRANCE
and
GdR 134 TdSI.

Abstract

In this paper, we show how data parallelism can be used for two low level vision algorithms based on deterministic relaxation techniques. First, we review the architecture and programming model of the CM2. Then, we give some details about the implementation. Finally, we compare both algorithms in terms of running time per iteration for different image sizes. We also present a comparison between parallel and serial implementations of the same algorithm on a CM2 and on a SUN4 respectively.

1 The Connection Machine

In this section, we briefly describe the architecture of the Connection Machine, a more detailed description can be found in [3],[5].The Connection Machine is a single instruction multiple data (SIMD) parallel computer with 8K to 64K processors. Each processor is a 1-bit serial processor, with 32K bytes of local memory and a 8MHz clock. The Connection Machine is accessed via a front end computer which sends macro-instructions to a microcontroller. All processors are cadenced by the microcontroller in receiving the same nano instruction at a given time from it. Physically, the architec ture is organized as followed:

- The CM2 Chip contains 16 1 bit processors.
- A Section is the basic unit of replication. It is composed of 2 CM2 Chips, the local memory of the 32 processors and the Floating Point Unit.
- The interprocessor communication architecture is composed of two distinct networks:
    - A nearest-neighbour network, the NEWS Network (North-East-West-South), interconnects Processors in groups of four.
    - A more complex network called the Router Network is used to provide general communication between any pair of processors. Each group of 16 processors is connected to the same router and each router is connected to 12 other routers forming a 12-dimensional hypercube.

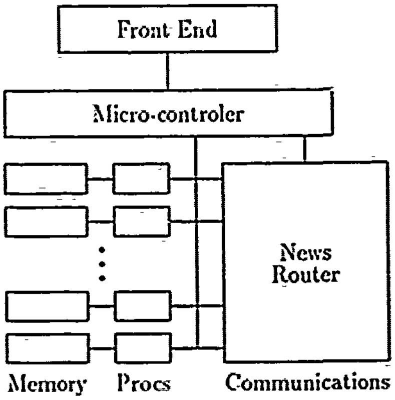


Figure 1: CM-2 architecture

For a given application, the user can dynamically define a particular geometry for the set of physical processors that has been attached.

The processor resource can be virtualized when the num ber of data elements to be processed is greater than the num ber of physical processors. In such a case, several data ele ments are processed on a single physical processor.
Such a data-parallelism model architecture is well suited to computer vision as it is expressed in [4], [6].

2 Parallel implementation of relaxation algorithms

2.1 Mathematical model

We use a probabilistic model of the image based on Markov Random Fields (MRF). Two fields are used to model the image: one for the intensity to be restored, the other one for the discontinuities (edges). The problem is the approximation of a surface given noisy depth data on a regular 2D lattice of sites. The value of the intensity field at each site (i.e. each pixel) is given by the surface height at that site.

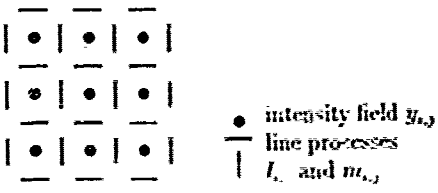


Figure 2: Image model

[0]The authors would like to thank D. Ray and H. Voorhees from Thinking Machine Corporation for many useful discussions

We suppose the noisy image is described by:

$$d_{i,j} = y_{i,j} + n_{i,j} \tag{1}$$

where $n_{i,j}$ is a white Gaussian noise and $y_{i,j}$ is the original data.

The energy can be expressed by:

$$E = D + S + P \tag{2}$$

where:

$$D = \sum_{i,j}(y_{i,j} - d_{i,j})^2 \tag{3}$$

$$\begin{aligned} S &= \sum_{i,j} \lambda^2((y_{i,j} - y_{i-1,j})^2(1 - l_{i,j}) \\ &\quad + (y_{i,j} - y_{i,j+1})^2(1 - m_{i,j})) \end{aligned}$$

which models the smoothing constraint ($\lambda^2 \star gradient^2$) and:

$$P = \sum_{i,j} \alpha(l_{i,j} + m_{i,j}) \tag{4}$$

which is the cost to pay for introducing an edge.

The problem is reduced to the minimization of a non-convex energy function. Usually two kinds of techniques are used to solve this problem.

- stochastic techniques such as Simulated Annealing,
- deterministic techniques such as Graduated Non Convexity, Mean Field Annealing, Iterated Conditional Mode.

We are interested in deterministic algorithms and their parallel implementation.

2.2 The algorithms

In this section we describe two deterministic algorithms for edge detection and image restoration: the graduated non convexity technique (GNC) originally proposed by Blake & Zisserman [1] and the mean field annealing (MFA) used by Geiger & Girosi [2] and Zerubia & Chellappa [7].

For the GNC the basic idea is the following:

- The first step is to build a convex approximation E^* of the energy E.
- The minimisation of E^* gives a global minimum.
- Then a sequence $E^{(p)}$ is built so that $E^1 = E^*$ and $E^0 = E$ for p=1, 1/2, 1/4, 1/8.
- the minimisation of $E^{(p)}$ uses as initial conditions the result of the minimisation of $E^{(p-1)}$

The MFA algorithm is based on Mean Field approximation used in statistical mechanics. This approximation consists of replacing the stochastic interaction among the fields at different locations by the interaction of the field at each site with the mean field values at different locations. Once the partition function Z has been approximated, it is easy to derive a set of deterministic equations for the mean field values $\bar{y}$, $\bar{l}$ and $\bar{m}$ of the intensity and the line-processes.

2.3 Parallel implementation

Both methods are based on the weak membrane model (cf.(2)) and both algorithms are inherently serial: each step produces a pixel map which is taken as input for the next step. For the GNC, we implement a checkerboard version of the successive over relaxation method [1] to minimize the energy and for the MFA, we use an optimal step conjugate gradient descent [7].

Although both techniques are deterministic, it takes a lot of computational time on a sequential computer to get the edge map and the restored image. Our attempt in reducing this time is based on the fact that, in early vision processing, much of the time is spent in performing local computation. Herein, we use data parallelism (one pixel per virtual processor) and fast local communications (NEWS) provided by the underlying architecture[3], [5]. For global operations like computing the energy value over the whole image, we use the reduce primitives.

3 Experimental results

These two algorithms have been run on a great variety of images. The discussion that follows is about their running time on the CM2 using 8K processors. Qualitative aspects are discussed in [8].

Table 1 shows the number of different types of instructions for each algorithm. All these instructions are 32 bit floating point instructions. The interesting point is that the interprocessor communications are local and the ratio between the number of communication instructions and the number of others is very low.

	Arith.and Compar.	NEWS	Global Ops.
GNC	117	10	1
MFA	123	24	4

Table 1 : Number of CM instructions per iteration.

The GNC and MFA parallel algorithms implemented on the Connection Machine are compared. For each one, we give:

- The Virtual Processor Ratio (VPR) i.e. the number of virtual processor per physical processor.

- The CM time.

- The total time.

- The number of iterations for each algorithm.

- The CM time per iteration.

In table 2 and 3 are displayed the results for two images: a noisy aerial image (cf.fig.3) and an infra-red image.

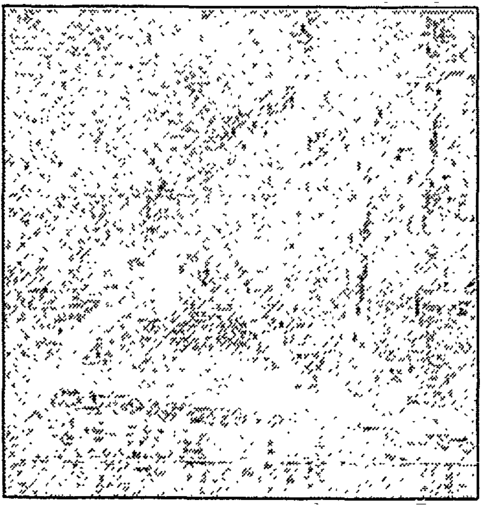


Figure 3: Noisy aerial image 128x128, (SNR=5db)

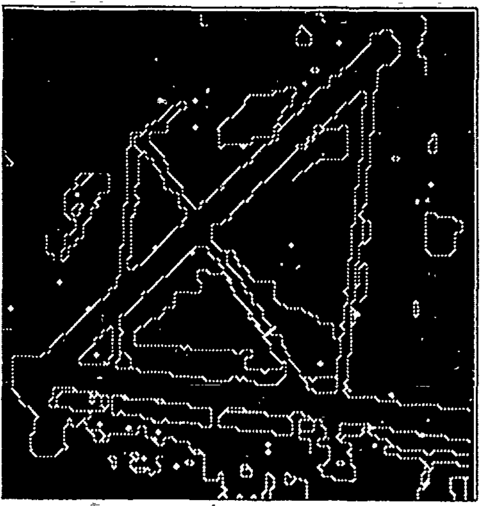


Figure 4: Edge map with GNC

	VPR	CM time	Total time	Nb.Iter	CM time per It
GNC	2	21	64.5	567	0.03
MFA	2	10.2	19.13	174	0.05

Table 2 : Noisy Aerial Image 128x128

	VPR	CM time	Total time	Nb.Iter	CM time per It
GNC	8	45.84	58.85	172	0.12
MFA	8	16.57	20.17	98	0.17

Table 3 : Infra-red Image 256x256.

The execution time per iteration for the GNC algorithm is faster than for the MFA. But, generally, the MFA needs less iterations to converge. Figure 5 shows that the execution time per iteration is proportional to the number of pixels per processor. However, as the number of pixels per processor grows the ratio between the execution-time per iteration and the number of pixels per processor is a bit lower. This is due to the fact that, the higher is the VPR, the higher is the percentage of local communications.

	Fortran-Sun4		*Lisp-CM	
	Total time	Time per it.	Total time	Time per it.
GNC	15mn.	1.69s	64.5s	0.11s
MFA	25mn.	7.97s	19.13s	0.109s

Table 4 : Noisy Aerial Image 128x128

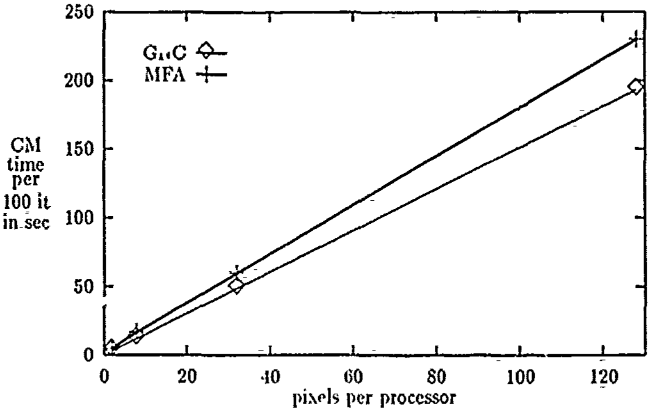


Figure 5: Execution time of the GNC and MFA Algorithms

4 REFERENCES

1. A. Blake and A. Zisserman, "Visual reconstruction", *MIT Press, Cambridge - MA*, 1987.
2. D. Geiger and F. Girosi. "Parallel and deterministic algorithms for MRFs : surface reconstruction and integration", *Proc. ECCV90*, Antibes, Apr. 1990.
3. W.D. Hillis, "The Connection Machine ", *Cambridge, MA, MIT Press*, 1985.
4. J. Little, G. Belloch and T. Cass, " Algorithmic techniques for computer vision on a fine-grain parallel machine", *IEEE Trans. on PAMI*, Vol. 11, pp 244-257, Mar. 1989.
5. Thinking Machine Corporation, "Connection Machine, Model CM2 Technical Summary" *TMC, Cambridge, MA*, May 1989.
6. H. Voorhees, D. Fritzsche and L. Tucker, "Exploiting data parallelism in Vision on the Connection Machine system", *Proc. 10th ICPR*, Atlantic City, Jun. 1990.
7. J. Zerubia and R. Chellappa, "Mean field approximation using Compound Gauss-Markov Random Field for edge detection and image restoration", *Proc. ICASSP90*, Albuquerque, Apr. 1990.
8. J. Zerubia and F. Ployette, "Edge detection and image restoration using two deterministic relaxation algorithms. Implementation on the Connection Machine CM2", *INRIA research report no.1291*, Oct.1990.

A FLEXIBLE MULTIPROCESSOR ARCHITECTURE FOR HIGH-LEVEL COMPUTATION IN ELECTRONIC MEASUREMENTS

Alessandro GANDELLI Vincenzo PIURI

Department of Electrical Engineering Department of Electronics
Politecnico di Milano
piazza L. da Vinci 32, I20133 Milano, Italy

ABSTRACT - The massive computation requirements of an increasing number of applications in electronic measurements need dedicated hardware architectures to match the high throughputs of real-time environments. This paper presents some results about design and implementation of an integrated measurement system based upon a multiprocessor architecture.

I. INTRODUCTION

A number of high-computing applications are nowadays practically feasible and appealing due to the availability of realisable, reliable structures for massive computation at reasonable costs, e.g. in digital signal processing. In particular, it is possible to implement specialised architectures for management of large data in real time for extracting a reduced amount of essential information. Complex and accurate data observation and system control may be implemented by using this characteristic figures of the input data. A great number of examples may be found in electrical and electronic areas: one of the most important applications that becomes now realisable is constituted by the systems for real-time measurement for power and armonic control in power distribution.

In this paper we present a new integrated approach to real-time measurement of electrical parameters which is based upon a multiprocessor system. Such architecture allows to satisfy the massive throughputs and requirements of complex, high-level measurements. In particular, we present the overall hardware architecture and the dedicated software environment for developing application-specific measurement algorithms.

Our parallel architecture is composed by a SIMD multiprocessor subsystem for parallel computation and by a front-end for an easy user interaction and control. Both such components are connected to a standard VME bus and to a dedicated control bus. The front-end computer is a standard Motorola's 68020 board, with mass-storage devices and input/output subsystems. Such computer provides the high-level human interface for designing, testing and executing the application-specific measurement algorithms. The front-end computer allows to load the measurement algorithm in the program memory of the multiprocessor subsystem, to overview and control the computation in such subsystem.

II. APPLICATION AND MATHEMATICAL REMARKS

Electronic measurements require execution of high-computing algorithms to extract basic characteristics and features from a large amount of data. Usually, when the measurements are used as information sources for chosing the suited actions in plant or process controls, all operations must be performed in real-time. This constraint makes impressive the amount of data that must be treated on-line during system management.

One of the most recent and important applications in the fields of electrical engineering is the real-time accurate control of high-voltage transmission and distribution systems. In this application, real-time electronic measurements are used to compensate the armonic contents of current and voltages due to the characteristics of the distribution system and to the electric loads. The basic aim of such operation is in fact the optimisation of power distribution with minimisation of power losses. Real-time control is important to measure dinamically the armonics and to provide continuously updated feedback information to the armonic compensator. This power electronic system suppresses unexpected and undesired frequencies and corrects reactive power.

High computational capabilities and possible parallelisation of the basic computation are two basic features to implement efficient architectures for real-time applications when massive computation is concerned [3] [4]. In the application case presented above, these charac teristics allow to guarantee a very low response time in driving the electric power system for armonic compensation and, thus, to achieve an effective and optimal power distribution.

In this application, there is also a very large set of input data, which are composed by the samples of three-phase voltages and currents Since control activities operate only on integral forms of such inputs, the high-level management of the system does not need a detailed and complete knowledge of the samples' values In fact analysis of the electrical behavior of the distribution system can be performed by extracting the characteristic figures both in time or frequency domains. A pre-processing and compacting function must therefore be executed to reduce the quantity of information This is useful to allow acquisition, manipulation and storing only of the necessary data in the control computer Data compression and reduction are an important step towards the implementation of real-time control of complex system at reasonable costs.

To satisfy strict computational requirements, the computer architecture must be able to guarantee a high computational power and, above all, an impressive input throughput Output is generally very smaller than input flow, due to the characteristics of the computation input/output management is thus strongly unbalanced towards input activities. Neither traditional computers nor known architectures of parallel processors [3] [4] are able to provide a flexible and high-throughput connection between the processing units and the external environment, at least at reasonable costs. Therefore, a new dedicated solution, as the one we propose, is needed to achieve an efficient system without using expensive structures.

By looking at the algorithms used in electronic measurement and in system controls, we can identify a low number of typical mathematical operations that are common to the most of the algorithms. Traditional arithmetic operations are obviously widely used in all numerical algorithms. Therefore, the architecture, specialised for measurements, must greatly optimise execution of data transfer, addition, subtraction, multiplication and division.

A second group of more complex operations constitutes the basic kernel of a large number of algorithms to compute electrical quantities: the data transformations. Different kinds of data transformations may be identified in the most of the popular algorithms. Some examples are Fourier, Walsh-Hadamard, sine and cosine transformations, convolution and correlation [5] [7]. These operations allow to extract the different spectral contents of the input data and to generate composed control signals. The previous mathematical operations require execution of many simple arithmetic operations (e.g. additions, subtractions and multiplications) [5], efficient generation of the corresponding coefficients and proper data addressing and use. An effective implementation should therefore be tailored to perform the most of such activities directly in hardware to increase the overall computational power of the architecture Configurable devices should be preferred to hard-wired solutions to guarantee the flexibility of the system and adaptability to different specific applications and measurements.

III. HARDWARE ARCHITECTURE

The parallel architecture, presented in this paper, has been designed especially for advanced measurement applications. Its features are thus strictly connected with the specific algorithms implemented in measurement procedures. The processor design has been studied to obtain the best performances not only for some traditional arithmetic operations, but also for high-speed digital signal processing operations (e.g. Walsh and Fourier transforms).

Our parallel architecture is composed by a *multiprocessor subsystem* and by a *front-end computer*, as it is shown in fig. 1. The first one allows to execute high-speed parallel computation, while the second one provides an easy user interaction and control. Highly-parallel computation is supported by using the 32-bit VME system bus and a 70-bit dedicated bus for instruction transfer and broadcasting. High throughputs are achieved by using dedicated 16-bit data busses (one for each processing unit) which connect the multiprocessor subsystem to the external data sources.

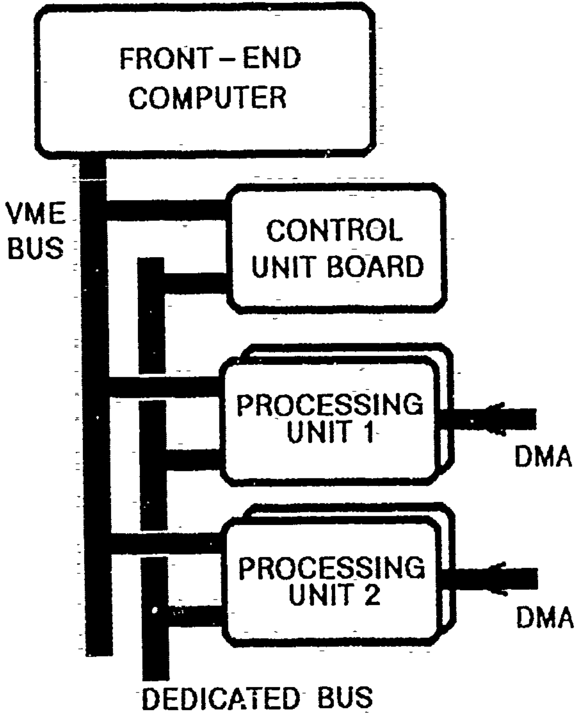


Fig. 1 - The multiprocessor architecture

Our structure differs from many other parallel structures since it behaves basically as a co-processing unit of the front-end computer. In fact it may be used to extract some characteristic figures from a large amount of input data by means of traditional signal processing algorithms. This pre-processing activity greatly reduces the quantity of data that must treated by the front-end computer, in particular for applications in electronic mesurements and automatic control.

The *front-end computer* is a standard Motorola's 68020 board, with hard and floppy disks, printer, plotter, serial and parallel communication channels, graphic display. The main goal of this computer is the definition of a standard Unix environment for the user activities. Moreover, it provides the high-level human interface for designing, testing and executing the application-specific measurement algorithms. In particular, the front-end computer has been designed to support loading of the measurement algorithms in the multiprocessor memory and controlling of the parallel computation. The multiprocessor subsystem and the front-end computer are connected to a standard VME bus and to a dedicated control bus for data and control exchange.

The *multiprocessor subsystem* is a SIMD architecture, in which a single instruction is performed at the same time by all processing units [3] [4]. The main goals of our design approach to a multiprocessor architecture were performance, modularity, expandability, flexibility and adaptability. Performance is an obvious requirement to implement structures for massive computation. Modularity allows to achieve a compact standard architecture in which the component can be modified and improved according to the requirements of the specific applications. Simple and fixed interfaces, interconnections and protocols guarantee the expandability of the structure (at least in a reasonable range) by addition of identical modules to match the algorithm characteristics. Flexibility and adaptability of the architecture to a large class of measurement algorithms is needed to implement a unique computing structure that may be used for a number of different applications Adaptability should hold also at run-time by means of a software reconfiguration of interconnections to guarantee a high availability and flexibility of the system Many of these features where achieved by adopting a simple, clear, high-speed, modular architecture based upon standard components.

The multiprocessor subsystem is composed by identical *processing units* and by a *controller unit* (see fig. 1). The processing units provide the parallel computational power by working on different data sets contemporaneously. The control unit has been introduced to overview the multiprocessor activity by imposing the sequence of instructions that must be performed by all processing units in parallel. The number of processing units depends on the specific measurement algorithm required by the user. We have found from extensive experiments that a good balance between the multiprocessor performance and host computer operativity in the minimal configuration can be achieved when the units are no more than 8.

Each *processing unit* is composed by two boards: the *ALU* (Arithmetic Lugic Unit) *board* and the *local memory*. The ALU board is the heart of the computational structure to implement high-speed algorithms for electronic measurement. The memory board is basically used to store measurement data coming from the external environment, during execution of the application-specific algorithm. Performant data transfer inside each board is guaranteed by an internal high-speed bus for data and addresses. Our implementation of the experimental multiprocessor, discussed in this paper, considers arithmetic units for 16-bit data and operates at 10 MHz to avoid hardware complex structures and to provide high flexibility, powerful computational capabilities and high-throughput data management at reasonable costs. The structure of the processing unit is given in fig. 2.

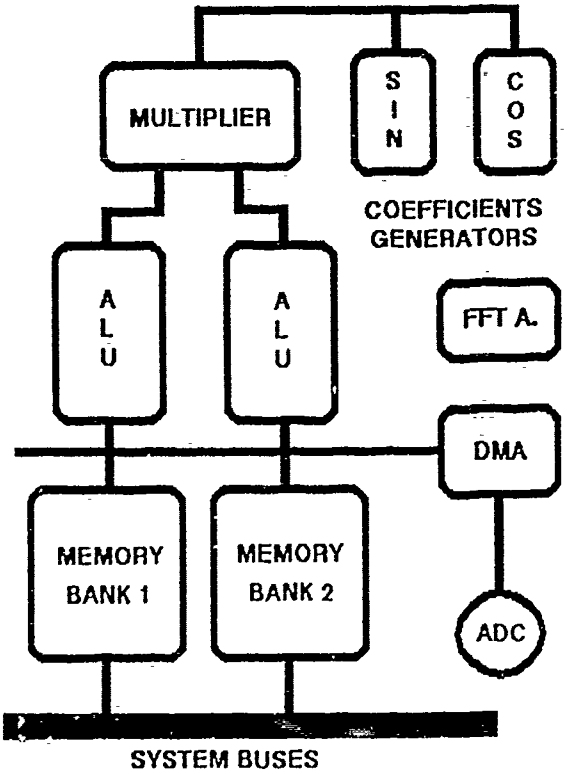


Fig. 2 - The processing unit

The *ALU board* is a custom structure for execution of basic arithmetic operation and some complex algorithmic kernels of digital signal processing. The ALU board is based upon AMD's 29501 ALU's and AMD's 29517 multiplier.

Two three-port VLSI ALU's perform the computation on 16 bit data using a multibus and multiregister architecture. A high performance 16x16 bit multiplier completes the arithmetic hardware and provide the suited computational support for a high-speed implementation of FFT algorithm. The two ALUs can perform multiple data operations during the same clock cycle using their multiport structure (two input-output ports and one input port) and the six register operating independently by the internal ALU and MUX registers

The *memory board* has been designed to obtain the best performance and to reduce data transfer time without limiting system capabilities. This board is strictly connected to ALU board Nevertheless, it has high autonomous capabilities for data management A double memory bank has in fact been developed to allow contemporary memory management from both internal processor resources and system ALU. Every bank is formed by four 8x32 kbit memories connected to the internal 32-bit bus. These two bus can be alternatively switched to one of the three possible connection paths (ALU's, direct I/O, system bus).

Two independent data paths are provided by the memory board in order to exchange information with the external environment.

- the main data path, which is connected to the system bus,
- the secondary data path, that provides a direct input bus for data acquisition by using the DMA technique.

While the processors are working on odd banks, the host computer performs data transfer operations from/to the main memory for the even banks, and vice versa. The switching system provides the control of active paths in order to avoid bus conflicts.

When DMA tecniques are adopted, at first the control logic executes memory reading operations (performed by the main CPU) and, then, DMA acquisition from secondary data port since data are written in the same storage. Both these activities can be implemented while the processor is working on the other bank because their execution time is smaller than time required to complete a Walsh or a Fourier transform.

The sine and cosine generators (for FFT algorithm) or Walsh function generator (for FWT algorithm) and latches providing the bus switching during computational activities have been arranged on the memory board to limit the ALU complexity.

As we have seen, the multi-access structure and management of input data and memory banks are the most important novelty of our approach to multiprocessor design for applications in electronic measurements. The variuos possibilities of overlapping data input and internal computations greatly increase the throughput and the computational power of the system to match the massive requirements of real-time applications in measurement and control. While one half of a processing unit is computing the nominal algorithm, the other is supporting the input/output mechanism.

Parallel computation and management of the boards of the processing units must be performed by activating the proper control signals on the dedicated control bus. Due to the complexity of the different activities that can be excuted in parallel and to the number of control signals, direct control from the front-end computer is not feasible.

To provide an efficient mechanism for operation fetching and decoding in the multiprocessor system, we defined a set of *instructions* that can be executed by our architecture. Each instruction corresponds is the compact coded representation of the control signals that must be activated to impose execution of the desired operation. A program for the multiprocessor system consists of a sequence of such instructions.

Since our multiprocessor is a SIMD machine, only one *control unit* is needed to supervise and manage all processing units. This board is composed by the *program memory* and the *processing controller*. The program memory contains the sequence of instructions (in the coded format) that the multiprocessor must execute. The processing controller performs the fetch and decode operations of the multiprocessor program.

The *processing controller* is an AMD's 2910 microcontroller with its own control circuits, private memory and latches. The microcontroller fetches and decodes the operations for the multiprocessor. For each new instruction, it acquires the coded representation from the program memory and identifies the current instruction. Then, it generates the proper control signals, which correspond to the current instruction, by looking at them into the its own private memory. Finally, the microcontroller moves to the next instruction. Due to the modularity of the multiprocessor architecture, in the second release of the control board, the microcontroller will be substituted by a microprocessor. when properly programmed such microprocessor will be able to generate autonomously sequences of multiprocessor instructions that will execute complex operations on data. In this case the controller unit will behave as a traditional microprogrammed unit.

The *program memory* is composed by twelve 8x4k bit high-speed static CMOS RAM. In this memory, we store the sequence instructions for the multiprocessor, which are 29-bit long. The memory is loaded by front-end computer through the system bus before compution is started. The 96-bit control signals, generated by the program controller in correspondence to each multiprocessor instruction, are imposed to every processing unit by using the dedicated control bus.

Program execution is started by the front-end computer. after loading the program in the program memory of the multiprocessor subsystem, the front-end computer enables the autonomous computation of the multiprocessor by stating the proper signals to the control unit Results computed by the multiprocessor are stored in the local memory of the multiprocessor itself The front-end computer may read them through the VME data bus and provide delivering to the user or storing in mass-storage devices.

IV. SOFTWARE ENVIRONMENT

The use of an advanced hardware architecture generally presents a number of practical difficulties, which reduce the usability of the system and a complete exploitation of the new features. In our architecture, this drawback is essentially related to the complexity of the structure and to the non-traditional parallel programming style. To avoid this drawback and to give access to new computational paradigms also to non-expert users, an integrated environment become necessary.

In our research we developed different tools which guarantee a high-level view to the hardware architecture and easy programming of a large class of numerical-intensive applications. A general schema of the software environment is shown in fig. 3. The main entities and tools we developed and integrated in the standard Unix environment of the fron-end computer are:

- the low level multiprocessor assembler and the related translator,
- the assembler run time library for direct access to all features of the system,
- the high-level language C and the related compiler,
- a high-level run-time library for traditional programming in C language,
- a program loader,
- a symbolic debugger for C language,
- an interactive control environment.

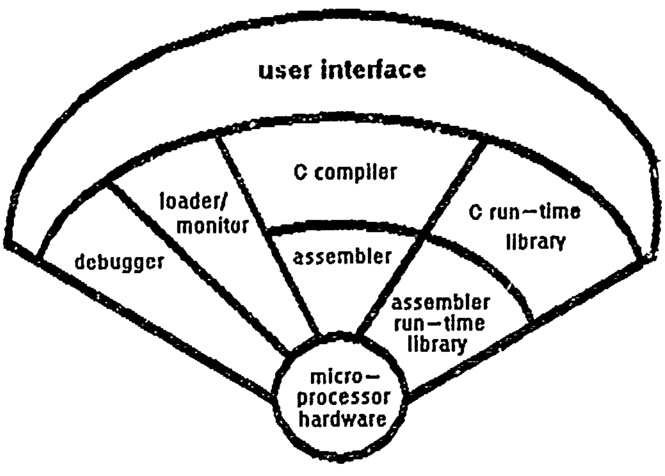


Fig. 3 - The software environment

To allow the user programming of application-specific algorithms for the multiprocessor architecture, it is necessary to define the sequences of instructions in the assembler of the multiprocessor itself. Such instructions are the coded representation of the control signals which impose execution of the desired sequences of operations. Whenever the multiprocessor controller fetches a new instruction, it decodes the compact representation and generates the proper signals to activate the devices required by the operation.

The symbolic representation of the assembler instructions of our multiprocessor is similar to all traditional assembler languages of microprocessors [1]. An instruction is given by the n-tuple

(opcode, op-1, op-2, ... op-n)

where *opcode* is the operator (i.e. the symbolic identifier of the desired operation), while *op-i* are the needed operands (i.e. immediate data, registers, memory addresses). The complete definition of the assembler language is given in [2]. On the front-end computer we developed a translator which generates the machine code from the assembler source code of the user program. Parallelism and data transfer are demanded to the proper design of the assembler program.

To write new application programs in the multiprocessor assembler in a simple way, we provide a modular, expandable run-time library. It contains the most used basic mathematical functions (e g. trigonometric, exponential and hiperbolic functions) and the advanced DSP operations (e.g. DFT, FFT, HFT, convolution, correlation). Such functions are coded as efficient as possible by exploiting all hardware parallelisms.

For loading, debugging and executing the machine code, the front-end provides a user-friendly environment. The user can select the program file that must be loaded, can transfer such program in the memory of the multiprocessor subsystem, can run the program Possibly, he can also execute the multiprocessor program in a step-by-step mode or by introducing break-points to observe the behavior of the system and to correct errors in algorithms under development.

A higher programming interface may be achieved by adopting a standard programming language and custom libraries for multiprocessor operations. In our system we consider at the moment a subset of the C standard programming language Control and programming of the multiprocessor architecture is allowed by a run-time library that we designed and implemented for the most common operations The user writes his programs as on a traditional monoprocessor computer during execution the run-time library guarantee the proper data transfer and program execution on the multiprocessor architecture. Program editing and compiling for the multiprocessor is performed onto the front-end computer by using a traditional text editor and our C cross-compiler, respectively. At the moment we are designing also a high-level debugging tool to provide a view of the program in the C language (except for the run-time multiprocessing procedures).

V. CONCLUDING REMARKS

A flexible multiprocessor architecture has been presented in this paper, particularly suited for on-line processing of a large amount of data signals The most original charateristics of such architecture is the high capability of acquiring data from the external environment due to specialised high-speed data paths for concurrent I/O operations. A powerful support to direct signal processing is the availability of dedicated hardware inside each processing unit for high-level mathematical computations (e.g. DFT). The traditional FFT alghorithm can be implemented in our architecture by using ten instructions they are iterately executed for a number of times equal to the number of butterflies required by the computation. If the number of sampled points is $N = 2^k$, the total execution time T for the FFT in a single processor board is $T = 10T_c kN/2 = 5T_c k2^k$, where T_c is the clock period of the system In our experimental implementation, a 1024-points FFT is executed about in 5 ms.

As software aspects are concerned, our system provides an integrated environment for development of application programs. A library of basic mathematical operation has been implemented. It is continuously updated and expanded to cope with the requiremets of new applications we are experimenting in the area of electronic measurements.

Further researches are presently in progress to improve the abstraction level for the programmer of applications. In particular, we are studying a user-friendly interface and programming style based upon icons: non-expert users will create the computational graph, which describes operations performed upon input data, through graphic interaction with the development environment. The user will select functional building blocks from a menu and connect them into the computational graph.

A second research topic is the automatic identification of parallelism in the user program. Compiler and assembler translator should be able to describe a sequence of simple operations affecting disjoint sets of devices in the same processing unit by means of a unique instruction. Whenever group of operations can be executed at the same time, the compiler and the assembler must detect this event and collapse the operations into the unique instruction.

REFERENCES

[1] Aho A.V., Ullman J.D., *Principles of Compiler Design*, Addison-Wesley, 1977

[2] Gandelli A., Piuri V., "EMMA - Electronic-Measurement Multiprocessor Assembler. definition and translator characteristics", Int. Rep. No. 025-90, Department of Electronics, Politecnico di Milano, 1990

[3] Hockney R.W., Jesshope C.R., *Parallel Computers*, Adam Hilger, Bristol, 1981

[4] Hwang K., Briggs F. A., *Computer Architecture and Parallel Processing*, McGraw-Hill, 1985

[5] Oppenheim, A.V., Schafer R.W., *Digital signal processing*, Prentice-Hall, NJ, USA, 1975

[6] Rose J., Loucks W., Vranesic Z., "FERMTOR: a tunable multiprocessor architecture", *IEEE Micro*, Vol. 5, Aug. 1985

[7] Van der Auweraer H., Snoeys R., "FFT Implementations Alternatives in Advanced Measurement System", *IEEE Micro*, Vol. 7, Feb. 1987

This work was supported by the CNR grant No. 90.00332.PF67

ON ASYMPTOTIC RELIABILITY FUNCTIONS FOR LARGE SERIES-PARALLEL SYSTEMS

KRZYSZTOF KOŁOWROCKI
Department of Mathematics
Maritime University
Gdynia, Poland

Abstract - A paper presents some theorems giving sufficient conditions for a reliability function to be an asymptotic of large series-parallel and parallel-series systems.

I. INTRODUCTION

We are interested in a wide class of systems which every two elements are connected to each other either parallel or series. In the investigation of these systems it can be noticed that their elements can be arranged and numerated this way that the system lifetime X is given by

$$X = \min_{i_1}\Big\{\max_{j_1} \ldots$$

$$\min_{i_m}\Big\{\max_{j_m}\{X_{i_1j_1\ldots i_mj_m}\}\Big\}\ldots\Big\}$$

or by

$$X = \max_{i_1}\Big\{\min_{j_1} \ldots$$

$$\max_{i_m}\Big\{\min_{j_m}\{X_{i_1j_1\ldots i_mj_m}\}\Big\}\ldots\Big\},$$

where

$$\mathfrak{X} = \Big(X_{i_1j_1\ldots i_mj_m} \;:\; i_1 = 1,2,\ldots,k,\ j_1 = 1,$$

$$2,\ldots,l_{i_1},\ldots,i_m = 1,2,\ldots,k^{(i_1j_1\ldots i_{m-1}j_{m-1})},$$

$$j_m = 1,2,\ldots,l_{i_m}^{(i_1j_1\ldots i_{m-1}j_{m-1})}\Big)$$

is the arranged family of the random variables corresponding with the lifetimes of the particular elements. In a first case the system is called a parallel-series system of order m and in a second case it is called a series-parallel system of order m, m ≥ 1. If for all $i_1, j_1, \ldots, i_m$

$$k^{(i_1j_1)} = k^{(i_1j_1i_2j_2)} = \ldots =$$

$$= k^{(i_1j_1\ldots i_{m-1}j_{m-1})} = k$$

and

$$l_{i_1} = l_{i_2}^{(i_1j_1)} = \ldots = l_{i_m}^{(i_1j_1\ldots i_{m-1}j_{m-1})} = l,$$

then parallel-series and series-parallel systems are called regular. Moreover, if random variables of the family $\mathfrak{X}$ have the same distribution function F(x), i.e. elements have the same reliability function R(x) = 1 - F(x), then these systems are called homogeneous.

Now, assuming $k = k_n$ and $l = l_n$, where n tends to infinity and k_n and l_n are natural numbers, we obtain sequences of parallel-series and series-parallel regular homogeneous systems of order m corresponding with the sequence (k_n, l_n). For these sequences of systems there exist sequences of reliability functions $\mathbb{R}_n^{(m)}(x)$ for parallel-series and $\bar{\mathbb{R}}_n^{(m)}(x)$ for series-parallel system.

II. ASYMPTOTICS OF PARALLEL-SERIES AND SERIES-PARALLEL SYSTEMS

Definition 1.
A reliability function $\mathbb{R}(x)$ is called an asymptotic reliability function of a sequence $\mathbb{R}_n^{(m)}(x)$ or an asymptotic of the regular homogeneous parallel-series system of order m if there exist constants $a_n^{(m)} > 0$, $b_n^{(m)} \in (-\infty, \infty)$, such that

$$\lim_{n\to\infty} \mathbb{R}_n^{(m)}(a_n^{(m)}x + b_n^{(m)}) = \mathbb{R}(x)$$

at all x where $\mathbb{R}(x)$ is continuous.

Similarly we define an asymptotic of the regular homogeneous series-parallel system of order m.

Theorem 1.

Lat

$$d = \sup\{x \;:\; \mathbb{R}(x) > 0\}.$$

If $\mathbb{R}(x)$ is continuous at point d in case when $d < \infty$ and sequences $(a_n^{(m)}, b_n^{(m)})$, (k_n, l_n) have the following properties

$$\lim_{n \to \infty} l_n^{m-1} k_n^{-\frac{1}{l_n}} = 0 \text{ for } m \geqslant 1$$

$$\lim_{n \to \infty} k_n^{l_n^{m-1} + \dots + 1} \left[F(a_n^{(m)} x + b_n^{(m)})\right]^{l_n^m} =$$

$$= -\ln \mathbb{R}(x) \text{ for } m \geqslant 1$$

at all $x \in (-\infty, d)$ where $\mathbb{R}(x)$ is continuous, then $\mathbb{R}(x)$ is an asymptotic of the regular homogeneous parallel-series system of order m for all $m \geqslant 1$.

Proof. ([4]).

Theorem 2.

Let

$$d = \inf\{x \;:\; \bar{\mathbb{R}}(x) < 1\}.$$

If $\bar{\mathbb{R}}(x)$ is continuous at point d in case when $d > -\infty$ and sequences $(a_n^{(m)}, b_n^{(m)})$, (k_n, l_n) have the following properties

$$\lim_{n \to \infty} l_n^{m-1} k_n^{-\frac{1}{l_n}} = 0 \text{ for } m \geqslant 1$$

$$\lim_{n \to \infty} k_n^{l_n^{m-1} + \dots + 1} \left[R(a_n^{(m)} x + b_n^{(m)})\right]^{l_n^m} =$$

$$= -\ln\left[1 - \bar{\mathbb{R}}(x)\right] \text{ for } m \geqslant 1$$

at all $x \in (d, \infty)$ where $\bar{\mathbb{R}}(x)$ is continuous, then $\bar{\mathbb{R}}(x)$ is an asymptotic of the regular homogeneous series-parallel system of order m for all $m \geqslant 1$.

Proof. ([4]).

III. CONCLUSION

Theorems 1 and 2 provide sufficient conditions for a reliability function to be an asymptotic of the regular homogeneous parallel-series and series-parallel systems respectively. These conditions allow to search for the possible asymptotics of the considered systems in case when their reliability models are fixed. First, one should assume any reliability function and next try to find a norming constants sequence $(a_n^{(m)}, b_n^{(m)})$ satysfying the sufficient conditions for the assumed reliability function to be an asymptotic of the system. Some examples and more general theorems about asymptotics of the parallel-series and series-parallel systems of order 1 can be found in [1,2,3,5]. Some examples of asymptotic for nonhomogeneous systems of any order m can be found in [4].

IV. REFERENCES

1. Barlow R. E., Proschan F., Statistical theory of reliability and life testing, Probability models, Holt, Rinehart and Winston, INC., New York, 1975.
2. Chernoff H., Teicher M., Limit distributions of the minimax of independent identically distributed random variables, Transactions of American Mathematical Society 116, 474-491, 1965.
3. Domsta J., Kołowrocki K., Examples of asymptotic reliability functions for large systems, Gdańsk University Scientific Journal, 32-49, Gdańsk, 1978.
4. Kołowrocki K., Reliability of series-parallel systems, Maritime University Scientific Journal, 97-136, Gdynia, 1981.
5. Kołowrocki K., Some remarks on a class of limit reliability functions for series-parallel systems, Maritime University Scientific Journal, 5-58, Gdynia, 1987.

AUTOMATIC 3-D HIERARCHICAL SUBSTRUCTURING SCHEME AND ITS IMPLEMENTATION ON A VECTOR-CONCURRENT MACHINE

MUKUL SAXENA
Corporate Research and Development
General Electric Company, Schenectady, N.Y. 12301

AND

RENATO PERUCCHIO
Department of Mechanical Engineering
University of Rochester, Rochester, N.Y. 14627

Abstract *This paper presents an extension to the use of generic algorithms for Recursive Spatial Decompositions (RSD) to design a hierarchical substruct jring (HS) scheme that can be easily coupled to the automatic mesh generator and can be embedded in a self-adaptive meshing-analysis procedure. The geometric algorithms for RSD generate nested-dissections (analytical substructures) of the domain that are used for the analysis scheme (HS) described in this paper. An evaluation of the HS scheme for parallel processing is discussed with reference to the results of implementation of the scheme on a vector-concurrent machine.*

1 INTRODUCTION

This paper describes a 3-D FEM analysis scheme based on Recursive Spatial Decompositions (RSD). The scheme forms an integral part of an automated FEM meshing-analysis system that is ideally suited for parallel computing [1]. Specifically, spatial-decompositions are used to (1) transform automatically a solid model into a finite element mesh, (2) perform incremental analysis via hierarchical substructuring, (3) produce hierarchical data structure that allows coupling between meshing and analysis, and (4) perform incremental self-adaptive analysis. The parallelism in the meshing scheme has been earlier described in [1]. In this paper the applicability of the hierarchical substructuring scheme is discussed in the context of parallel processing environments. A brief overview of RSD based meshing algorithm that forms the basis of the substructuring scheme is given in the following section.

2 AUTOMATIC MESHING SCHEME

A two-stage automatic meshing procedure is used for automatic generation of the finite-element meshes. For a detailed description see Reference [2]. In Stage 1, the solid to be meshed, Ω, is approximated by a collection of variably sized octants through the recursive spatial decomposition of the original geometric domain. Such an approximation is conveniently represented by a logical tree structure whose node have eight sons, popularly known as "octree". Stage 2 of the meshing algorithm transforms the RSD into a valid FEM mesh, through further processing of the individual octants.

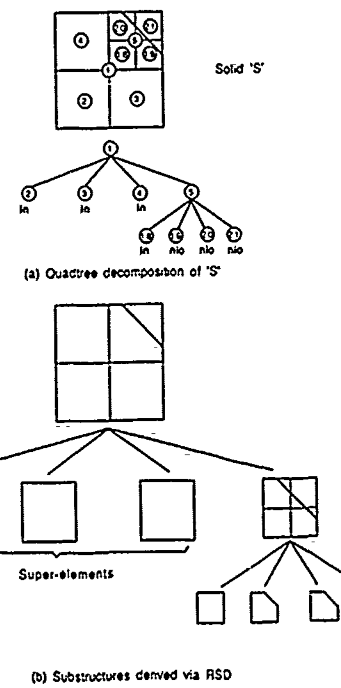
Each node of the octree represents an informationally complete subdomain ω_i and, in terms of the finite element model, a substructure. Thus, for the purpose of analysis, the octree can be regarded as a cataloging structure with geometrical and analytical information directly mapped on to it – see Figure 1 for a 2-D example. Such a substructuring scheme, whereby all the substructures are hierarchically organized and are derived through the recursive spatial decomposition (RSD) of the original domain, is referred to as *hierarchical substructuring.* Note that the analytical substructures produced by the RSD are equivalent to the *nested dissections* described in [3].

3 HIERARCHICAL SUBSTRUCTURING

There are three distinct stages in Hierarchical Substructuring.

In the first stage, stiffness matrices are formulated for the lowest level nodes of the octree which consist of assemblies of linear isoparametric elements. The stiffness matrix formulation is based on the degeneration of the eight-noded isoparametric brick element.

Second stage is the assembly stage of analysis. Starting from the bottom of the tree, the stiffness matrices of the offspring of the same-parent cell are assembled into a substructure and the interior degrees of freedom are eliminated by static condensation. Once all the substructures at a given level are assembled and condensed, the procedure begins to operate on the tree level immediately above.



(a) Quadtree decomposition of 'S'

(b) Substructures derived via RSD

Figure 1: *A 2-D structure and the associated substructures derived via RSD*

This process continues until the root node of the tree is reached. The mathematical formulation of this stage can be described as follows: Let b and i denote respectively the nodes on the boundary and the interior of the substructure. The equilibrium equations for the substructure can be partitioned as

$$\begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix} \begin{Bmatrix} X_i \\ X_b \end{Bmatrix} = \begin{Bmatrix} R_i \\ R_b \end{Bmatrix}, \tag{1}$$

where X is the displacement vector, R is the load vector, K_{ii} are the stiffness coefficients related to the interior nodes alone, K_{bb} the coefficients for the nodes on the boundary and K_{ib} the cross-coupling terms linking boundary nodes to the nodes in the interior. By eliminating the interior degrees of freedom a new system of linear equations is obtained:

$$K^r_{bb} X_b = R^r_b \tag{2}$$

where K^r_{bb} (= $K_{bb} - K_{bi}K^{-1}_{ii}K_{ib}$) and R^r_b (= $R_b - K_{bi}K^{-1}_{ii}R_i$) are the reduced stiffness matrix and the reduced load vectors, respectively.

The third stage involves root level solution and recovery of displacements and stresses. The reduced system of equations, obtained at the end of assembly stage, is solved for the unkown boundary displacements, K_{bb}. Once the root level displacements X_b are known the interior nodes are computed by solving the system

$$K_{ii} X_i = R_i - K_{ib} X_b. \tag{3}$$

Recall that K_{ii} has already been reduced to a upper triangular matrix during assembly stage. Thus X_i can be derived directly through backward substitution in Equation 3. Finally, the displacements at the element level are used to compute the stresses in the solid.

$$\sigma = DB\delta^e , \qquad (4)$$

where D is the material property matrix, B is the strain-displacement matrix and δ^e is the elemental displacement vector.

4 PARALLEL IMPLEMENTATION

Four domains were meshed using the octree-based mesh generator described in Section 2, and analyzed via hierarchical substructuring. Figure 2 shows the meshes used, the applied constraints and loads, as well as the deformed shape.

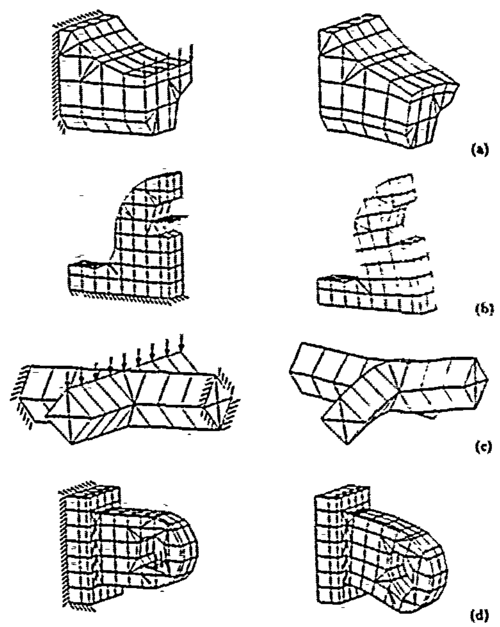


Figure 2: *Analysis via hierarchical substructuring – original meshes and deformed shapes for – (a)* block, *(b)* housing, *(c)* cyl_cyl_int, *and (d)* bracket.

To study parallelism, it is convenient to identify three stages of varying computational complexity as follows: 1) substructures assembly and condensation except the root level (this includes computation of element stiffness and condensation of all the substructures up to level 1 in the octree); 2) solution of the final root level substructure; and 3) recovery of interior d.o.f.'s. Stages 1) and 2) are the dominant factors in terms of solution time and, therefore, only these two stages are considered for parallel processing. Stage 2 of processing consists of assembling the eight level-1 substructures and then solving the final root-level substructure. As such, stage 2 must be performed on a single processor. However, since it requires the reduction of a large, nearly fully populated matrix, stage 2 can take full advantage of vectorization.

In order to minimize the computational overheads associated with parallel processing (processor idle time) the following strategy was used for mapping the computational load to various processors

1. Start at the lowest level of the octree and visit all the sub structures at this level.
2. Identify the first non-empty substructure S_i^* that can be assembled, i.e., all its sons are already assembled. (Hereafter, an asterisk denotes a substructure whose eight sons are already assembled.) Assign S_i^* to the first available processor P_1. The next non-empty substructure S_j^* is assigned to the second available processor P_2 and so on. Once all the eight sibling of an octree node are computed, the substructure corresponding to the parent node is marked as available for assembly.
3. The process continues until all the first level substructures are computed.

Table 1 shows the speed-ups and efficiencies for the example problems. ρ, η, T_S and T_P are the speed-up factor, efficiency, serial, and parallel time, respectively. All the times are reported for implementation on Alliant FX/8 machine configured for 8 processors. $\Delta\rho$ is the percent increase in the efficiency achieved by using the balanced processor load scheme, described above, in contrast to a scheme whereby eight level-1 substructures are assigned to separate processors. Notice that the increase in efficiency is marginal for the *bracket* problem. This can be ascribed to the fact that for this particular problem, condensation of one substructure dominates the whole execution cycle. Therefore, the processing time is only marginally influenced by the way the other substructures are assigned to different processors.

Object	T_S (sec)	T_P (sec)	ρ	η (%)	$\Delta\rho$ (%)
Block	9.19	1.222	7.52	94.0	41.0
Housing	23.30	6.154	3.79	47.4	45.0
Cyl_cyl_int	12.74	1.727	7.38	92.3	85.0
Bracket	85.67	30.853	2.78	34.8	30.0

Table 1: *Speed-up and efficiency for balanced load scheme*

Problem	T_{scalar} (sec)	T_{vector} (sec)	ρ
Block	12.123	5.613	2.16
Housing	56.866	13.475	4.22
Cyl_cyl_int	3.560	1.650	2.15
Bracket	270.064	46.565	5.79

Table 2: *Speed-ups due to vectorization of root level solution on the ALLIANT FX/8*

Table 2 reports the speed-ups produced by exploiting vectorization for solving the root level substructure. Notice that the efficiency of vectorization increases with the length of the associated vectors (i.e. the size of the stiffness matrix), as illustrated by the speed-ups for *housing* and *bracket* problem.

5 SUMMARY

This paper has presented a view that emphasizes on the use of RSD for 3-D Hierarchical Substructuring (HS) scheme that can be incorporated in a parallel FEM analysis system consisting of automatic meshing, analysis via HS, and self-adpative incremental re-meshing and re-analysis. The octree structure provides a tight coupling between geometrical and analytical data, thus allowing for an efficient integrated meshing-analysis procedure [1]. In conclusion, the recursive formulation of the algorithms makes the scheme ideally suited for parallel processing.

References

1 M. Saxena, "Parallel Algorithms for 3-D Automatic Meshing and Hierarchical Substructuring", PhD Dissertation, Department of Mechanical Engineering, University of Rochester, Rochester, New York, 1989.

2 R. Perucchio, M. Saxena, and A. Kela, "Automatic mesh generation based on recursive spatial decompositions of solids", *International Journal for Numerical Methods in Engineering*, vol. 28, pp. 2469-2501, 1989.

3 A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, Inc., New Jersey, 1981

A Parallel Algorithm for Clique Finding Problem*

Chain-Wu Lee
Institute of Computer Science and Information Engineering
National Chiao-Tung University
Hsinchu, Taiwan, R.O.C.

Shian-Shyong Tseng
Institute of Computer and Information Science
National Chiao-Tung University
Hsinchu, Taiwan, R.O.C.

ABSTRACT

In this paper, we try to solve clique finding problem, one of the most famous *NP-Complete Problems*, in parallel. We propose a new sequential clique finding algorithm, the empirical and theoretical performances of the algorithm are further analyzed. The empirical results show that the average time complexity is about $k \cdot e^{0.148n}$, which grows in an exponential order. However, this algorithm can be easily parallelized if an MIMD shared-memory machine is used and then a parallel clique finding algorithm is obtained. Therefore, the time complexity of our parallel algorithm is about $k \cdot e^{0\ 148n}/n$.

1 Clique Finding Problem and Algorithm

The clique finding problem is defined as follows.

Definition. **Given a graph $G = (V,E)$, the clique finding problem is to find every subset $V' \subseteq V$, such that every two vertices in vertices V' are joined by an edge in E [3].**

Clique finding problem is one of the most famous *NP-Complete problems* Generally speaking, exponential time is needed for it, however, there are some special cases which can be solved in polynomial [2] [4] [5] [6] [7].

Next, we define some notations and terminologies used in this paper. Degree of a vertex v is $d(v)$; $A(v)$ is the adjacent vertices of vertex v whereas $AE(v)$ is the adjacent edges of vertex v. Clique $k(G)$ is a set of vertices which are mutually connected in the graph G, $K(G)$ is a set of cliques in the graph G and all of the cliques in G constitute of a set called $K_{all}(G)$. $k_v(G)$ is a clique which $v \in k_v(G)$; likewise, $K_v(G)$ is the set of cliques that all cliques in this set include vertex v. The output of our algorithm is composed of two parts, essential part and optional part. Every vertex in the essential part is a clique and each combination of the optional part forms a clique with essential part.

If we give every vertex v in graph G a label, then $\lambda(v)$ is the integer value label of the vertex v. $G_v^*(V^*, E^*)$ is said to be a *restricted induced subgraph* with respect to vertex v in graph G iff $G_v^*(V^*, E^*)$ is an induced subgraph of G and (1) $V^* = v \cup A(v) - \{v' | v' \in A(v) \text{ and } \lambda(v') < \lambda(v)\}$. (2) $E^* = \{(u,w) | u, w \in V' \text{ and } (u,w) \in E\}$.

Now the notion of our algorithm is stated as below .

Given a graph G, let $k(G) = \{v_1, v_2, \cdots, v_m\}$ be a clique of G , $1 \leq m \leq n$ where n is the number of vertices in G. It is trivial that in graph G, for any arbitrary clique k, either $v \in k$ or $v \notin k$ This idea can be extended to a set of cliques, that is, for a vertex v, we have $K_{all} = K_v \bigcup K_{\bar{v}}$, where $K_v = \{k | v \in k\}$ and $K_{\bar{v}} = \{k | v \notin k\}$.

Second, if a vertex v is connected to all vertices in graph G, then the cliques $K_v(G)$ may be either $\{v\}$ or $\{k | k = k_c \cup \{v\}, k_c \in k(G')\}$, where G' is derived from G by removing v and its adjacent edges. So we have the following lemma.

Lemma 1 . **If v is a vertex which connects to all vertices in graph $G(V,E)$, then $K_v(G) = \{\{v\}, \{k | k = c \cup \{v\}, c \in k(G')\}\}$ where $G' = (V - v, E - AE(v))$.**

From Lemma 1, if v connects to all vertices in G_v^* then $K_v(G_v^*)$ must be either $\{v\}$ or $\{v\} \cup K(G_v^*(V^* - v, E^* - AE(v)))$.

If the vertices of the graph G are labelled with positive integers, then we have the following lemma.

Lemma 2 . **$\forall u \in k_v, u \neq v$ if $\lambda(v) < \lambda(u)$ then $k_v \subseteq V^*$ where V^* is the vertices of G_v^*.**

Proof : Suppose the lemma is false. That is, $k_v \nsubseteq V^*$. Then there must exist at least one vertex $w \in k_v, (v,w) \in E$ and $w \notin V^*$, such that either $\lambda(w) > \lambda(v)$ or $\lambda(w) < \lambda(v)$. For the former, it contradicts the definition of G_v^* because that if a vertex $w \neq v, (v,w) \in E$ and $\lambda(w) > \lambda(v)$, then w must be a vertex of G_v^*, that is $k_v \subseteq V^*$. For the latter, it contradicts the prerequisite of the lemma since $\lambda(v)$ is the smallest number among all vertices in k_v. Thus w can not exist in k_v. ■

According to Lemmas 1 and 2, we have Theorem 1.

Theorem 1 . $K(G) = \bigcup K_v(G_v^*)$.

Proof : Assume $K(G) = \bigcup K_v(G_v^*) + K'$, and assume for $k' \in K'$, $k' = \{u_1, u_2, \cdots, u_n\}$ and u_1 is with the smallest label among $\lambda(u_i)$. According to Lemma 2, we have $k' \in G_{u_1}^*$, contradicting to our assumption. ■

This theorem, in other words, means that the cliques in graph G can be found by discovering all the cliques in the restricted induced subgraphs of every vertex in graph G.

Next step, a parallel clique finding algorithm based on MIMD shared-memory machine with n processors is demonstrated.

For an n vertices graph, initially the processor PE_i of the MIMD shared-memory machine holds a vertex v_j with $\lambda(v_j) = i$, and grasps its restricted induced subgraph $G_{v_j}^*$ independently. The $K(G_{v_j}^*)$ can be obtained by unifying $\{v_j\}$ and the cliques in $G'(V^* - v_j, E^* - AE(v_j))$. So in the first step, we add v_j to a clique list. The clique list is a list whose elements are the members of a clique. Besides, it constructs the essential part of clique.

Next let's take a look at $G'(V^* - v_j, E^* - AE(v_j))$. If this subgraph is a clique, then any combination of this subgraph (optional part) can form a clique with the clique list. The exploration does not need further extend and the entire procedure can be terminated If the subgraph is not a clique, the processor PE_i first picks up a vertex, say v_k, compute $G_{v_k}^*(V^* - v_k, E^* - AE(v_k))$, push $G''(V^* - v_j - v_k, E^* - AE(v_j) - AE(v_k))$ into memory pool, and exploit the algorithm recursively to find the solutions.

If a processor PE_i finishes the job, the subgraph pushed into the memory pool previously can be dispatched to the free processor PE_i and repeats entire process.

The output of our algorithm contains two parts - an essential and an optional part. For the essential part, it is easy and straightforward, just output it. However, for the optional part, if we need to know *every* clique in the graph, we may adopt any arbitrary parallel combination generating algorithm on it and combine the output of the combination generating algorithm together with essential part[1].

In the previous paragraph, we have not pointed out the labeling methodology used in our algorithm, but sometimes the labeling scheme may dramatically affect the load of the processor and the performance of the algorithm.

Thus we proposed the following labeling scheme. Label every vertex v according to its degree $d(v)$ in ascending order, the smaller the degree of a vertex is, the smaller it labels.

By applying our algorithm and labeling scheme, we prune the smaller cliques of the graph and left the largest clique not processed. This concept is based on the "Largest clique is the most time consuming part in the graph in computation".

2 Time Complexity Analysis of the Clique Finding Algorithm

In the following section, we try to analyze the time complexity of the parallel clique finding algorithm. Note that under the MIMD model when any processor is free, we assume it gets a job and no communication exists between any two subtasks. So we may suppose the speedup of the parallel algorithm is n relative to its sequential version. Therefore , if the time complexity of sequential algorithm be denoted as $E_n(p)$ then the time complexity of parallel algorithm is certainly $E_n(p)/n$ for an n processors machine.

The judgment criteria of the algorithm here we define is the number of cliques output. Therefore, according to our algorithm for random labeling scheme, first step a vertex X is picked out for computation of the restricted induced subgraph. Suppose the degree of the picked vertex X is x. Then the probability of the restricted induced subgraph

G_X^* is $\binom{n-1}{x} p^x (1-p)^{n-1-x}$.

For the restricted induced subgraphs G_X^*, a clique is output and then the vertex X is deleted from it The rest restricted induced subgraphs will be recursively exploited by the algorithm, so the time steps required for this algorithm is $E_x(p)$ for the rest restricted induced subgraph; therefore the time complexity for G_X^* part is

$$\binom{n-1}{x} p^x (1-p)^{n-1-x} E_x(p) + 1$$

where 1 is the time used for the essential part.

*This research was partially sponsored by the National Science Council under NSC79 - 0108 - E009 - 17

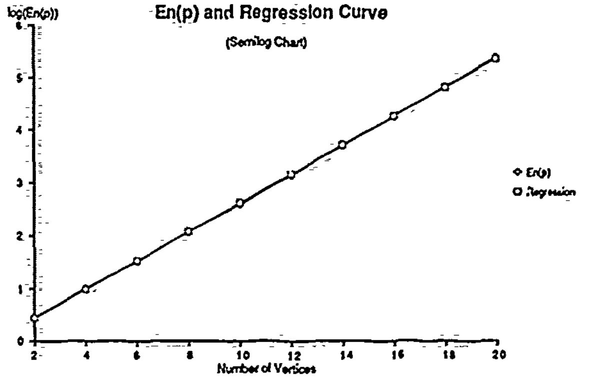


Figure 1: E_n and Regression Curves, Semilog chart

Note that x may vary from 0 to $n-1$, so the time complexity for n from 0 to $n-1$ is

$$\sum_{x=0}^{n-1}\left(\begin{array}{c} n-1 \\ x \end{array}\right)p^x(1-p)^{n-1-x}E_x(p)+1$$

After computing G_X^* part, the graph $G'(V-X, E-AE(X))$ obviously contains $n-1$ vertices, so the time needed is $E_{n-1}(p)$. Therefore the total time complexity is

$$E_n(p)=\sum_{x=0}^{n-1}\left(\begin{array}{c} n-1 \\ x \end{array}\right)p^x(1-p)^{n-1-x}\cdot E_x(p)+1+E_{n-1}(p)$$

And solve the above equation, we may get

$$E_n(p)=n+\sum_{k=0}^{n-2}(g_n(k+1)[\prod_{m=1}^{k}(1+g_n(m))](n-(k+1)))+ g_n(n-1)\cdot\prod_{m=1}^{n-2}(1+g_n(m))$$

From semi-log chart (see Fig. 1), we know that $E_n \approx ke^{0.27n}$.

3 Simulation Results

Since the performance analysis of our algorithm with sorted degree labeling strategy is quite hard, therefore a set of simulation is held in the last section.

In the simulation, first, for every probability p and number of vertices n, the experiment iterates 129 times. The range of n is from 1 to 25 step 2 whereas p ranges from 0.00 to 1.00. Graph representation for the simulation results is shown in Fig. 2. Semi-log graph for Fig. 2 is listed in Fig. 3.

After carefully examining the figure, the curve seems to be a straight line when $n \geq 5$, so the technique of linear regression is used to approximate this curve. Approximate line is also shown in Fig. 3. Undoubtedly, fitted curve is very proximate to the original curve. So the average time used in the our algorithm is about $\log(E_n) \approx 0.148 * n + c$ where E_n is average time used for our algorithm, n is the number of vertices, c is a constant and 0.148 is the slope of the straight line.

Expanding the formula, we get $E_n \approx k \cdot e^{0.148n}$.

4 Discussions

Maximum Clique Problem is to find a clique k where $|k| \geq |k'|$, $k' \in K_{all}$. This problem is an *NP-Complete problem*.

A slight modification on our algorithm will be more efficient for the maximum clique problem. For sequential algorithm, first, we generate all restricted induced subgraph for the graph. Say, $G^*_{v(1)}, G^*_{v(2)}, \quad, G^*_{v(n)}$ if there are n vertices in the graph; since maximum clique is the clique with greatest number of vertices in the graph, next step computation starts with $G^*_{v(x)}$ where the number of vertices of $G^*_{v(x)}$ is the largest among $G^*_{v(1)}, G^*_{v(2)}, \cdots, G^*_{v(n)}$, the above procedure will be repeatedly executed until the maximum clique is found. During the procedure we shall keep track of the size of vertices in the restricted induced subgraph generated Once found the size of restricted induced subgraph generated is smaller than any other restricted induced subgraph G^*_Q, then the procedure jumps to process G^*_Q. This algorithm is somewhat similar to the breadth first search.

For parallel algorithm, we should keep a location for maintaining the current maximum clique size $|k|$, and every processor gets a restricted induced subgraph with the same number $|k|$ of vertices and then repeat the sequential algorithm.

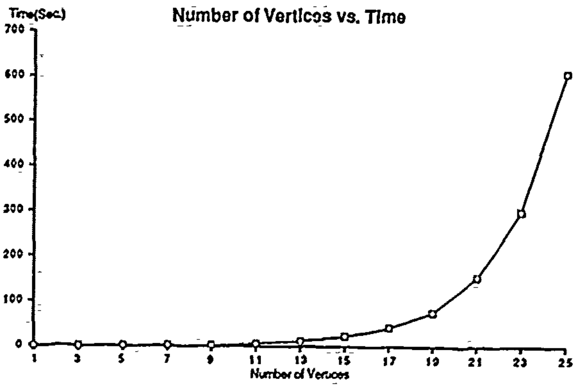


Figure 2: Number of vertices vs. average time

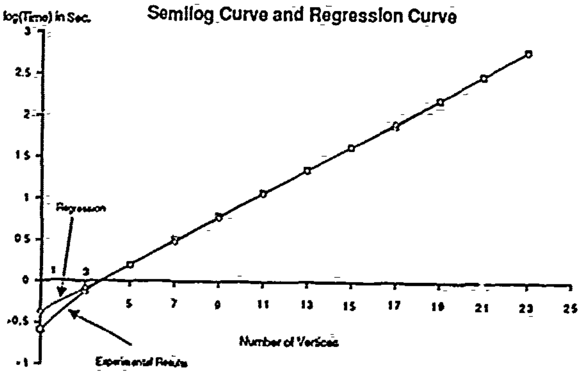


Figure 3: Semilog Chart vs. Linear Regression Approximation

References

[1] S G Akl, The Design and Analysis of Parallel Algorithms, Prentice-Hall, New Jersey, 1989.

[2] C Bron. J Kerbosch, "Finding All Cliques of an Undirect Graph". C. ACM, 19 (1973), pp.575-577.

[3] M. Gary and D. Johnson, Computers and Intractability : A Guide to the Theory of NP-Completeness, Freeman and Co., San Francisco, 1979.

[4] F. Gavril, "Algorithm for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph", SIAM J. Comput., 1 (1972), pp. 180-187.

[5] F Gavril, "Algorithm for a Maximum Clique and a Maximum Independent Set of a Circle Graph", Network, 3 (1973), pp. 261-273.

[6] H C Johnston, "Cliques of a Graph - Variations on the Bron-Kerbosch Algorithm", International Journal of Computer and Information Sciences, 5 (1976), pp.209- 238.

[7] C. H. Papadimitriou and M Yannakakis, "The Clique Problem for Planar Graphs" . Information Processing Letters, 13 (1981), pp. 131- 133.

NEW ALGORITHMS FOR POLYNOMIAL AND TRIGONOMETRIC INTERPOLATION ON PARALLEL COMPUTERS

ILAN BAR-ON AND AVRAM SIDI
DEPARTMENT OF COMPUTER SCIENCE, TECHNION,
TECHNION CITY, HAIFA 32 000, ISRAEL.

Abstract.
An interpolation polynomial of order N is constructed from p independent subpolynomials of order $n \sim N/p$ Each such subpolynomial is found independently and in parallel. Moreover, evalluation of the polynomial at any given point is done independently and in parallel, except for a final step of summation of p elements. Hence, the algorithm has almost no communication overhead and can be implemented easily on any parallel computer. We give examples of finite-differcnce interpolation, trigonometric interpolation, Chebyshev interpolation, and conclude with the general Hermite interpolation problem.

1. Introduction. In this paper we study the problem of polynomial and trigonometric interpolation on large parallel MIMD computers. There are well known sequential methods for both problems. However, these methods are not easily adaptable to parallel systems, and especially to loosely connected systems such as rings, stars, because of the overhead due to interprocessor communication, see for example[2, 3, 5]

In this work we present new algorithms for polynomial and trigonometric interpolation that require almost no communication between the processors. Given an interpolation problem of order $N = np$, p being the number of processors, we . . divide it into p smaller interpolation problems of order n. T se problems are solved independently and in parallel using .n appropriate sequential method. The value of the interpolation polynomial is then a combination of the corresponding subvalues.

2. The interpolation polynomial. Let $f(x)$ be a function defined on $[a, b]$, whose values on the set of $N+1$ distinct points $X = \{x_0, x_1, \ldots, x_N\}$, is given by $f_j \equiv f(x_j), j = 0, 1, \ldots, N$. We are interested in constructing a representation of the polynomial $P(x)$ of degree at most N that interpolates $f(x)$ on X and is most suitable for parallel computation. Let $\{X_1, X_2, \ldots, X_p\}$ be a partition of X,

$$X = \cup_1^p X_i, \quad \text{and} \quad X_i \cap X_j = \Phi, \ i \neq j.$$

The following theorem indicates how $P(x)$ can be constructed independently and in parallel by p processors, each solving a smaller interpolation problem on one of the subsets X_i.

THEOREM 2.1. *For $i = 1, \ldots, p$, define*

$$w_{i,j}^{-1} = \prod_{x_k \notin X_i} (x_j - x_k), \quad x_j \in X_i,$$

and let $Q_i(x)$ be the polynomial of degree at most $|X_i| - 1$ that satisfies the following interpolation conditions:

$$Q_i(x_j) = w_{i,j} f_j, \quad x_j \in X_i. \tag{1}$$

Then $P(x)$, the interpolation polynomial on X, is given by

$$P(x) = \sum_{i=1}^{p} Q_i(x) \prod_{x_k \notin X_i} (x - x_k). \tag{2}$$

Proof. See [1]. □

It is known[4, 6, 7] that the barycentric representation for Lagrange interpolation enjoys a large d. ;ree of numerical stability. We, therefore, look for a generalization of the barycentric representation that is appropriate for the formula given in (2).

THEOREM 2.2. *Let $Q_i(x)$ be as in Theorem (2.1), and let $R_i(x), i = 1, \ldots, p$, be the polynomial of degree at most $|X_i| - 1$ that satisfies the interpolation conditions*

$$R_i(x_j) = w_{i,j}, \quad x_j \in X_i. \tag{3}$$

Then $P(x)$ can be express.d in the form

$$P(x) = \frac{\sum_{i=1}^{p} Q_i(x) / \prod_{x_i \in X_i} (x - x_i)}{\sum_{i=1}^{p} R_i(x) / \prod_{x_i \in X_i} (x - x_i)} \equiv \frac{\sum_{i=1}^{p} \phi_i(x)}{\sum_{i=1}^{p} \psi_i(x)}. \tag{4}$$

Proof. See[1]. □

Given p processors, we assign processor $i = 1, \ldots, p$, to computing the corresponding terms $\phi_i(x)$ and $\psi_i(x)$. The computation of the $w_{i,j}$ takes $O(n_i(N - n_i))$ additions and multiplications in the worst case, where $n_i = |X_i|$. However, as will be seen in the following sections, in many cases of interest these values can be computed analytically in much fewer operations. Assuming that the $w_{i,j}$ are known, and that $n_i \sim n \sim N/p, i = 1, \ldots, p$, each processor is faced with an interpolation problem of order n that can be solved in parallel with no need of interprocessor communication. Once the interpolation polynomial is known, its value at points not in the set are given by summing and dividing th. corresponding subvalues in (4).

In Sections 3,4,5 we consider the problems of finite difference interpolation, trigonometric interpolation, and Chebyshev interpolation. For ease of representation we will use a slightly different notation as follows: We assume that the function $f(x)$ is given at $N = np$ distinct points and that each of the p subsets X_i, which are now numbered by $i = 0, \ldots, p-1$, contains exactly n points. We denote the points in the subset X_i by $x_{i,j}$, $j = 0, \ldots, n-1$.

3. Finite Difference Interpolation. Let X be a set of equally spaced points in the interval $[a, b]$,

$$x_i = a + ih, \quad h = \frac{b-a}{N-1}, \quad \text{for } i = 0, 1, \ldots, N-1.$$

We assume for simplicity that $N = np$ and p is the number of processors available. We consider here two partitions.

In the first partition we assign the ith group of n consecutive points to the ith processor,

$$X_i = \{x_{i,j} = x_{in+j}, \quad j = 0, \ldots, n-1\},$$

for $i = 0, \ldots, p-1$. Hence,

$$\begin{aligned} w_{i,j}^{-1} &= \prod_{k=0}^{in-1} (x_{in+j} - x_k) \prod_{k=(i+1)n}^{N-1} (x_{in+j} - x_k) \\ &= C_1 (-1)^{in} \binom{n-1}{j} / \binom{N-1}{in+j}, \end{aligned}$$

where $C_1 = (-h)^{N-n}(N-1)!/(n-1)!$ is independent of i and j. We note that if the same constant C multiplies all the $w_{i,j}$, it follows from (1),(3) that the subpolynomials $Q_i(x)$ and $R_i(x)$ are also multiplied by the same constant C but the interpolation polynomial $P_n(x)$ remains invariant by (4). In view of this, each processor has to compute the corresponding polynomials $Q_i(x)$ and $R_i(x)$ that satisfy the interpolation conditions

$$\begin{aligned} Q_i(x_{i,j}) &= (-1)^{in} f_{in+j} \binom{N-1}{in+j} \Big/ \binom{n-1}{j}, \\ R_i(x_{i,j}) &= (-1)^{in} \binom{N-1}{in+j} \Big/ \binom{n-1}{j}, \end{aligned}$$

for $j = 0, \ldots, n-1$, and this computation can be carried out using any finite difference formula.

In the second partition the subsets X_i are formed according to

$$X_i = \{x_{i,j} = x_{i+jp}, \quad j = 0, \ldots, n-1\},$$

for $i = 0, 1, \ldots, p-1$. Consequently,

$$\begin{aligned} w_{i,j}^{-1} &= \frac{\prod_{k=0}^{i+jp-1}(x_{i+jp} - x_k) \prod_{k=i+jp+1}^{N-1}(x_{i+jp} - x_k)}{\prod_{l=0}^{j-1}(x_{i+jp} - x_{i+lp}) \prod_{l=j+1}^{n-1}(x_{i+jp} - x_{i+lp})} \\ &= C_2(-1)^{i+j(p-1)} \binom{n-1}{j} \Big/ \binom{N-1}{i+jp}, \end{aligned}$$

where $C_2 = C_1/p^{n-1}$ is independent of i and j. Each processor has to compute the corresponding polynomials $Q_i(x)$ and $R_i(x)$ that satisfy the interpolation conditions

$$\begin{aligned} Q_i(x_{i,j}) &= (-1)^{i+j(p-1)} f_{i+jp} \binom{N-1}{i+jp} \Big/ \binom{n-1}{j}, \\ R_i(x_{i,j}) &= (-1)^{i+j(p-1)} \binom{N-1}{i+jp} \Big/ \binom{n-1}{j}, \end{aligned}$$

for $j = 0, \ldots, n-1$. As before, this computation can be carried out using any finite difference formula.

We have the following operation count for constructing and evaluating the polynomial, when $p \ll N$:

	sequential	parallel
Construction	$(N-1)N/2$	$(n-1)n$
Evaluation	N	$2(n+p)$

We obtain a speed up of order $p^2/2$ in the construction stage, and a speed-up of order $p/2$ in the evaluation stage as compared to the ordinary sequential finite difference methods.

4. Trigonometric interpolation. Let θ_j, be equally spaced points in $[0, 2\pi]$ given by

$$\theta_j = \frac{2\pi j}{N}, \quad j = 0, 1, \ldots, N-1,$$

and let $f(\theta)$ be a function defined on $[0, 2\pi]$ whose values $f_j \equiv f(\theta_j), j = 0, 1, \ldots, N-1$, are given. Furthermore, let $N = 2M$. Then there exists a unique balanced trigonometric polynomial $T(\theta)$ of degree M,

$$T(\theta) = \frac{1}{2}a_0 + \sum_{k=1}^{M-1}(a_k \cos k\theta + b_k \sin k\theta) + \frac{1}{2}a_M \cos M\theta \quad (5)$$

interpolating $f(\theta)$ at the points $\theta_j, j = 0, 1, \ldots, N-1$, see[4]. A complex interpretation of $T(\theta)$ in terms of the variable $z = e^{i\theta}$ yields the balanced complex trigonometric polynomial $P(z) \equiv T(\theta)$ of degree M,

$$P(z) = \frac{1}{2}c_{-M}z^{-M} + \sum_{k=-M+1}^{M-1} c_k z^k + \frac{1}{2}c_M z^M,$$

with $c_{-M} = c_M$, whose coefficients c_k are related to the a_k and b_k in (5) through

$$a_k = c_k + c_{-k}, \quad b_k = i(c_k - c_{-k}), \quad k = 0, 1, \ldots, M.$$

Of course, $P(z)$ satisfies the interpolation conditions

$$P(z_j) = T(\theta_j) = f_j, \quad z_j = z_1^j, \; z_1 = e^{i2\pi/N}. \qquad (6)$$

for $j = 0, 1, \ldots, N-1$, and the coefficients c_l,

$$c_l = \frac{1}{N}\sum_{k=0}^{N-1} f_k z_k^{-l}, \quad l = -M, -M+1, \ldots, M,$$

can be computed in $O(N\log N)$ operations using the FFT. In this section we introduce a new representation for $P(z)$ that can be evaluated in parallel using the FFT on smaller sets of points. Let $N = np$ with $n = 2m$, and consider the partition $\{Z_0, Z_1, \ldots, Z_{p-1}\}$ of the set of points $Z = \{z_0, z_1, \ldots, z_{N-1}\}$, where
$Z_l = \{z_{l,r} = z_{l+rp} = z_l z_p^r, \; r = 0, 1, \ldots, n-1\}, \quad l = 0, 1, \ldots, p-1.$

THEOREM 4.1. *For $l = 0, 1, \ldots, p-1$, let*

$$w_{l,r}^{-1} = z_{l,r}^{-M+m} \prod_{k \neq l}(z_{l,r} - z_{k,s}), \quad r = 0, 1, \ldots, n-1,$$

and let $Q_l(s)$ be the balanced complex trigonometric polynomial of degree m that satisfies the interpolation conditions

$$Q_l(z_p^r) = f_{l+rp} w_{l,r}, \quad r = 0, 1, \ldots, n-1,$$

on the subset of points Z_l. Then $P(z)$, the balanced trigonometric polynomial that satisfies the interpolation conditions in (6), can be expressed in the form

$$P(z) = z^{-M+m}\sum_{l=0}^{p-1} Q_l(z/z_l) \prod_{k \neq l}(z - z_{k,s})$$

Proof. See[1]. □

As before, we look for a generalized barycentric formula. This formula is developed in Theorem 4.2 below.

THEOREM 4.2. *For $l = 0, 1, \ldots, p-1$, let $\hat{Q}_l(s)$, be the balanced complex trigonometric polynomial of degree m that satisfies the interpolation conditions*

$$\hat{Q}_l(z_p^r) = (-1)^{r(p-1)} f_{l+rp}, \quad r = 0, 1, \ldots, n-1,$$

on the subset of points Z_l. Then the balanced trigonometric interpolation polynomial $P(z)$ of Theorem (4.1) can be expressed in the form

$$P(z) = \frac{\sum_{l=0}^{p-1}(-1)^l z_l^{-m} \hat{Q}_l(z/z_l)/((z/z^l)^n - 1)}{\sum_{l=0}^{p-1}(-1)^l z_l^{-m} \hat{R}(z/z_l)/((z/z^l)^n - 1)},$$

where, $\hat{R}(s)$ assumes the simple forms

$$\hat{R}(s) = \begin{cases} 1 & \text{if } p \text{ is odd,} \\ \frac{1}{2}(s^m + s^{-m}) & \text{if } p \text{ is even.} \end{cases}$$

Proof. See[1]. □

Now that we have obtained the barycentric form of $P(z)$, we can obtain that of $T(\theta)$, the real form of $P(z)$, very easily as follows:

$$T(\theta) = \frac{\sum_{l=0}^{p-1}(-1)^l \hat{U}_l(\theta-\theta_l)/\sin m(\theta-\theta_l)}{\sum_{l=0}^{p-1}(-1)^l \hat{V}_l(\theta-\theta_l)/\sin m(\theta-\theta_l)},$$

where $\hat{U}_l(\phi) \equiv \hat{Q}_l(s)$, with $s = e^{\mathrm{i}\phi}$, is the balanced trigonometric polynomial of degree m that satisfies the interpolation conditions

$$\hat{U}_l(\theta_{pr}) = (-1)^{r(p-1)} f_{l+rp}, \quad r = 0, 1, \ldots, n-1,$$

and $\hat{V}(\phi)$ is given by

$$\hat{V}(\phi) = \begin{cases} 1 & \text{if } p \text{ is odd.} \\ \cos m\phi & \text{if } p \text{ is even.} \end{cases}$$

We have the following operation count for constructing and evaluating the polynomial, when $p \ll N$

	sequential	parallel
Construction	*Nlog N*	*nlog n*
Evaluation	N	$n+2p$

We obtain a speed-up of order p both in the construction and evaluation of the polynomial as compared to the sequential FFT algorithm.

5. Chebyshev interpolation.

Let x_j, be N Chebyshev points

$$x_j = \cos\theta_j, \quad \theta_j = \frac{2j+1}{2N}\pi, \; j = 0, 1, \ldots, N-1,$$

in $[-1, 1]$. Let, $f(x)$ be a function defined on $[-1, 1]$ whose values $f_j \equiv f(x_j), j = 0, 1, \ldots, N-1$, are given. Let $N = np$, and consider the partition $X = \{X_0, X_1, \ldots, X_{p-1}\}$, where

$$X_l = \{x_{l,r} = x_{l+rp}, \; r = 0, 1, \ldots, n-1\}, \quad l = 0, 1, \ldots, p-1.$$

We define a new partition, $Y = \{Y_0, Y_1, \ldots, Y_{q-1}\}$, $q = \lfloor (p+1)/2 \rfloor$ by

$$Y_l = X_l \cup X_{l'}, \quad l = 0, 1, \ldots, q-1, \; l' = p-1-l.$$

LEMMA 5.1. *For $l = 0, 1, \ldots, q-1$, define*

$$w_{l,r}^{-1} = \prod_{k \neq l,l'} (x_{l,r} - x_{k,t}), \quad r = 0, 1, \ldots, n-1,$$

and let $Q_l(x)$ be the polynomial of degree $|Y_l| - 1$ that satisfies the interpolation conditions

$$Q_l(x_{k,r}) = f_{k+rp} w_{k,r}, \quad k = l, l', \; r = 0, 1, \ldots, n-1,$$

on the subset of points Y_l. Then $P(x)$, the interpolation polynomial on X, can be expressed in the form

$$P(x) = \sum_{l=0}^{q-1} Q_l(x) \prod_{k \neq l,l'} (x - x_{k,t}). \tag{7}$$

Proof. The result in (7) follows from Theorem 2.1. □

In developing the barycentric formula we distinguish between the cases in which p is even and odd.

THEOREM 5.2. *Let p be even, and for $l = 0, 1, \ldots, q-1$, let $\hat{Q}_l(x)$, be the polynomial of degree $2n-1$ that satisfies the interpolation conditions*

$$\hat{Q}_l(x_{k,r}) = f_{k+rp}, \quad k = l, l', \; r = 0, 1, \ldots, n-1.$$

Then $P(x)$ of Lemma 5.1 can be expressed in the form

$$P(x) = \frac{\sum_{l=0}^{q-1}(-1)^l \sin 2n\theta_l \hat{Q}_l(x)/(T_{2n}(x) - T_{2n}(x_l))}{\sum_{l=0}^{q-1}(-1)^l \sin 2n\theta_l/(T_{2n}(x) - T_{2n}(x_l))},$$

where $T_{2n}(x)$ is the Chebyshev polynomial of the first kind.

Proof. See[1]. □

THEOREM 5.3. *Let p be odd, and for $l = 0, 1, \ldots, q-2$, let $\hat{Q}_l(x)$, be the polynomial of degıɛ $2n-1$ that satisfies the interpolation conditions*

$$\hat{Q}_l(x_{k,r}) = \begin{cases} (-1)^r f_{l+rp}, & k = l \\ (-1)^{r+1} f_{l'+rp}, & k = l' \end{cases} \quad r = 0, 1, \ldots, n-1.$$

Furthermore, let $\hat{Q}_{q-1}(x)$, be the polynomial of degree $n-1$ that satisfies the interpolation conditions

$$\hat{Q}_{q-1}(x) = f_{q-1+rp}, \quad r = 0, 1, \ldots, n-1.$$

Then $P(x)$ of Lemma 5.1 can be expressed in the form

$$P(x) = \frac{\sum_{l=0}^{q-2} \frac{(-1)^l \sin 2n\theta_l \hat{Q}_l(x)}{T_{2n}(x) - T_{2n}(x_l)} + \frac{(-1)^{q-1}\hat{Q}_{q-1}(x)}{2T_n(x)}}{\sum_{l=0}^{q-2} \frac{(-1)^l 2\sin n\theta_l T_n(x)}{T_{2n}(x) - T_{2n}(x_l)} + \frac{(-1)^{q-1}}{2T_n(x)}}. \tag{8}$$

Proof. See[1]. □

We next show how to find the corresponding polynomials $\hat{Q}_l(x), l = 0, 1, \ldots, q-1$ using the FFT algorithm. We give explicit formulas for the case where p is odd. The case where p is even is solved similarly. Let $Q(x)$ be a polynomial of of degree $m-1$, and let its representation in terms of the Chebyshev polynomials of order less than m be

$$Q(x) = \frac{1}{2}a_0 + \sum_{k=1}^{m-1} a_k T_k(x).$$

Rewriting the series in terms of $x = \cos\theta$, $z = e^{\mathrm{i}\theta}$, we get the corresponding complex Chebyshev polynomial of degree $m-1$

$$C(z) = \sum_{k=-m+1}^{m-1} c_k z^k, \quad c_k = c_{-k} = \frac{1}{2}a_k, \; k = 0, 1, \ldots, m-1.$$

Let $Q(x)$ staisfies the interpolation conditions

$$Q(x_j) = g_j, \quad x_j = \cos\theta_j, \; \theta_j = \frac{2j+1}{2m}, \; j = 0, 1, \ldots, m-1.$$

Then $C(z)$ satisfies the interpolation conditions

$$C(v_j) = P(x_j) = g_j, \quad v_j = e^{\mathrm{i}\theta_j},$$

and vice versa. Hence, $Q(x)$ can be obtained from $C(z)$.

THEOREM 5.4. *Let $\hat{C}(s)$ be the balanced complex trigonometric polynomial of degree n that satisfies the interpolation conditions*

$$\hat{C}(z_1^j) = \frac{1}{2} f_{q-1+jp}, \quad \hat{C}(z_1^{n+j}) = \frac{1}{2} f_{q-1+j'p}$$

for $j = 0, 1, \ldots, n-1$, where $j' = n-1-j$ and $z_1 = e^{\mathrm{i}2\pi/2n}$.

Then $C(z)$, the complex Chebyshev polynomial of degree $n-1$ corresponding to $\hat{Q}_{q-1}(x)$ in (8), can be expressed in the form

$$C(z) = \hat{C}(z/z_1^{1/2}) + \hat{C}(1/(zz_1^{1/2}))$$

Proof. See[1]. □

THEOREM 5.5. *Let $\hat{C}(s)$ be the balanced complex trigonometric polynomial of degree n that satisfies the interpolation conditions*

$$\hat{C}(z_1^j) = f_{l+jp}, \quad \hat{C}(z_1^{n+j}) = f_{l'+j'p}$$

for $j = 0,1,\ldots,n-1$, where $j' = n-1-j$,, $z_1 = e^{i2\pi/2n}$ and $l < q-1$. Then $C(z)$, the complex Chebyshev polynomial of degree $2n-1$ corresponding to $(-2i\sin 2n\theta_l\hat{Q}_l(x))$ in (8), can be expressed in the form

$$C(z) = ((\frac{z}{v_{l'}})^n - (\frac{z}{v_{l'}})^{-n})\hat{C}(\frac{z}{v_l}) + ((\frac{1}{zv_{l'}})^n - (\frac{1}{zv_{l'}})^{-n})\hat{C}(\frac{1}{zv_l})$$

Proof. See[1]. □

Again we obtain a speed-up of order p both for the construction and evaluation of the polynomial when $N \sim n(2p), p \ll N$,

	sequential	parallel
Construction	$Nlog\,N$	$2nlog(2n)$
Evaluation	N	$2(n+p)$

6. The general Hermite interpolation problem. Let $M+1$ distinct points $X = \{x_0, x_1, \ldots, x_M\}$, in the interval $[a,b]$, be given and let $f(x)$ be a function defined on $[a,b]$, for which

$$f_j^t \equiv f^{(t)}(x_j), \quad t = 0,1,\ldots,k_j-1, \; j = 0,1,\ldots,M.$$

We are interested in constructing a representation of the general Hermite interpolation polynomial $P(x)$ of degree at most N, $N+1 = \sum_{j=0}^{M} k_j$, that interpolates $f(x)$ on X, i.e.,

$$P^{(t)}(x_j) = f_j^t, \quad t = 0,1,\ldots,k_j-1, \; j = 0,1,\ldots,M,$$

and is most suitable for parallel computation.

Let $\{X_1, X_2, \ldots, X_p\}$ be a partition of X,

$$X = \cup_1^p X_i, \quad X_i \cap X_j = \Phi \text{ for } i \neq j.$$

The following theorem indicates how $P(x)$ can be constructed independently and in parallel by p processors, each solving a smaller general Hermite interpolation problem on one of the subsets X_i.

THEOREM 6.1. *For $i = 1,\ldots,p$, and $x_j \in X_i$, define*

$$q_{i,j}^t = (\frac{f_j^t}{t!} - \sum_{s=0}^{t-1} q_{i,j}^s v_{i,j}^{t-s})/v_{i,j}^0, \quad t = 0,1,\ldots,k_j-1.$$

where

$$v_{i,j}^0 = \Pi_{x_r \notin X_i}(x_j - x_r)^{k_r}, \quad v_{i,j}^t = \frac{1}{t}\sum_{s=0}^{t-1} v_{i,j}^s z_{i,j}^{t-1-s},$$

for $t = 1,2,\ldots,k_{j-1}$, and

$$z_{i,j}^{t-1} = (-1)^{t-1} \sum_{x_r \notin X_i} \frac{k_r}{(x_j - x_r)^t}.$$

Let $Q_i(x)$ be the polynomial of degree at most $n_i - 1$, $n_i = \sum_{x_j \in X_i} k_j$, that satisfies the following interpolation conditions.

$$Q_i^{(t)}(x_j) = t!q_{i,j}^t, \quad t = 0,1,\ldots,k_j-1.$$

Then $P(x)$, the interpolation polynomial on X, is given by

$$P(x) = \sum_{i=1}^{p} Q_i(x)\Pi_{x_r \notin X_i}(x - x_r)^{k_r} \equiv \sum_{i=1}^{p} Q_i(x) l_i(x).$$

Proof. See[1]. □

THEOREM 6.2. *The general Hermite interpolation polynomial $P(x)$ of Theorem 6.1 has the barycentric form*

$$P(x) = \frac{\sum_{i=1}^{p} Q_i(x)/\prod_{x_j \in X_i}(x - x_j)^{k_j}}{\sum_{i=1}^{p} R_i(x)/\prod_{x_j \in X_i}(x - x_j)^{k_j}}$$

where $R_i(x)$, like $Q_i(x)$, is a polynomial of degree at most $n_i - 1$ that satisfies the same interpolation conditions with f_j replaced by 1, and f_j^t by 0, $t = 1,\ldots,k_j-1$, for all j.

Proof. As in Theorem 2.2. □

We can find the formulas for $v_{i,j}^s$, $s = 0,1,\ldots,k_j-1$, in

$$O(\sum_{x_j \in X_i} k_j|X - X_i| + k_j^2) \leq O(n_i|X - X_i| + n_i^2)$$

operations, and the formulas for the $q_{i,j}^s$, $s = 0,1,\ldots,k_j$, in

$$O(\sum_{x_j \in X_i} k_j^2) \leq O(n_i^2)$$

operations. Let $n_i \sim n \sim N/p, i = 1,\ldots,p$, and assume that the $v_{i,j}^s$ are known. Each processor is then faced with a general Hermite interpolation problem of order n, that can be solved in $O(n^2)$ operations.

7. Conclusion. We have presented a new interpolation polynomial that is especially useful for parallel computers as its construction and evaluation requires almost no communication between the processors. The interpolation problem is divided into smaller independent subproblems that can be solved independently using any known sequential interpolation method. Thus we have reduced the problem from order N to order $n \sim N/p$. Furthermore, we have developed a barycentric formula that enjoys a high degree of numerical stability as in the case with the barycentric formula for the ordinary Lagrange interpolation.

REFERENCES

[1] I. BAR-ON AND A. SIDI, *New algorithms for polynomial and trigonometric interpolation on parallel computers*, Tech. Report 660, Technion, Computer Science Department, 1990.

[2] M. L. DOWLING, *A fast parallel Horner algorithm*, SIAM J. Comput., 19 (1990), pp. 133-142.

[3] O. EĞECIOĞLU AND E. GALLOPOULOS, *A parallel method for fast and practical high-order Newton interpolation*, BIT, 30 (1990), pp. 268-288.

[4] P HENRICI, *Essentials of numerical analysis*, John Wiley & Sons, 1982.

[5] J. REIF, *Logarithmic depth for algebraic functions*, SIAM J. Comput., 15 (1986), pp. 231-242.

[6] W. J. TAYLOR, *Method of lagrangian curvilinear interpolation*, J. Research National Bureau of Standards, 35 (1945), pp. 151-155.

[7] W. WERNER, *Polynomial interpolation. Lagrange versus Newton*, Mathematics of Computation, 43 (1984), pp. 205-217.

A NEW DIVIDE AND CONQUER PARALLEL ALGORITHM FOR THE CHOLESKY DECOMPOSITION OF BAND MATRICES

ILAN BAR-ON
DEPARTMENT OF COMPUTER SCIENCE, TECHNION,
TECHNION CITY, HAIFA 32 000, ISRAEL.

Abstract.

We present a new divide and conquer parallel algorithm for finding the Cholesky decomposition of a band symmetric positive definite matrix. This is the first time such an algorithm is presented. All previously known parallel algorithms for this problem are direct implementations of the sequential methods, which as though, offer almost no speedup. Here, for the first time a parallel oriented algorithm is presented, with an approximate speedup of order $p/3$ given p processors. Moreover, the algorithm can be implemented on many existing parallel computers.

We further discuss the more theoretical aspects of the algorithm, and show that it can be implemented in $O(log\ m\ log\ n)$ time using $p = (n/m)\mathcal{M}(m)/(log\ m\ log\ n)$ processors. Here, $\mathcal{M}(m) = m^\beta, 2 \leq \beta \leq 3$, and $m^\beta/log\ m$ denotes the least number of processors required in order to multiply two matrices of order m in $O(log\ m)$ time. This improves by a factor of $log\ m$ the best previously known result for this problem.

We conclude with an application of the algorithm to the finding of the eigenvalues of a non-singular band symmetric matrix. We show for the first time how to implement each iteration of the QR algorithm in the same complexity as above.

1. Introduction. In this paper we study the problem of finding the Cholesky decomposition of a band symmetric positive definite(s.p.d.) matrix on large parallel MIMD computers. Such a decomposition is important in many numerical methods for solving systems of linear equations and for computing the eigenvalues and eigenvectors of a corresponding matrix.

All previously known parallel algorithm for this problem are direct implementations of the sequential methods, see for example [4, 6, 7]. As such, they suffer from the inherent sequential nature of these methods and in most interesting cases, where the bandwidth is small, they offer almost no speed-up.

In this work we present a new divide and conquer parallel algorithm which offers an approximate $p/3$ speedup over known sequential methods, given p processors. The problem of finding the Cholesky decomposition of a band matrix of order n is reduced to the of finding p independent decompositions of small band matrices of order n/p. Each such problem is then solved independently and in parallel.

2. The algorithm. Let A be a band s.p.d. matrix of order $n = qm$ and bandwidth m, i.e.,

$$A = \begin{pmatrix} A_1 & L_1 & & & \\ U_1 & A_2 & L_2 & & \\ & \ddots & \ddots & \ddots & \\ & & U_{q-2} & A_{q-1} & L_{q-1} \\ & & & U_{q-1} & A_q \end{pmatrix}, \quad \begin{matrix} U_i, A_i, L_i \in M(m), \\ U_i = L_i^t \text{ is} \\ \text{upper triangular.} \end{matrix} \tag{1}$$

We assume for simplicity that $q = lp$, where $p = 2^{k+1}$ is the number of processors. Let $A^s, s = 0, 1, \ldots, k$ denotes a block structuring

$$A^s = \begin{pmatrix} A_1^s & L_1^s & & & \\ U_1^s & A_2^s & L_2^s & & \\ & \ddots & \ddots & \ddots & \\ & & U_{v-2}^s & A_{v-1}^s & L_{v-1}^s \\ & & & U_{v-1}^s & A_v^s \end{pmatrix}, \quad \begin{matrix} v = p/2^s\ s = 0, 1, \ldots, k, \\ U_i^s, A_i^s, L_i^s \in M(2^s n/p), \\ A_i^s \textit{ band s.p.d.} \end{matrix} \tag{2}$$

Here, $A_i^s, i = 1, \ldots, v$ denotes the ith principal submatrix of order $2^s n/p$, i.e., rows and columns $(i-1)2^s n/p+1, \ldots, i2^s n/p$, of A The offdiagonal block elements L_i^s, U_i^s are given by

$$L_i^s = \begin{pmatrix} 0 & 0 \\ L_{il2^s} & 0 \end{pmatrix} = \begin{pmatrix} \widetilde{L_i^s} & 0 \end{pmatrix}, \quad \widetilde{L_i^s} = \begin{pmatrix} 0 \\ L_{il2^s} \end{pmatrix}, \tag{3}$$

$$U_i^s = \begin{pmatrix} 0 & U_{il2^s} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \widetilde{U_i^s} \end{pmatrix}, \quad \widetilde{U_i^s} = \begin{pmatrix} U_{il2^s} \\ 0 \end{pmatrix}, \tag{4}$$

where $\widetilde{L_i^s}, \widetilde{U_i^s} \in M(2^s n/p \times m)$.

Let $E_i^s, F_i^s, G_i^s, H_i^s$, matrices of order m, be given by

$$A_i^s X_i^s = \widetilde{L_i^s}, \quad X_i^s = \begin{pmatrix} E_i^s \\ \vdots \\ F_i^s \end{pmatrix}, \quad \begin{matrix} s = 0 \ldots k, \\ i = 1, 3 \ldots p/2^s - 1. \end{matrix} \tag{5}$$

$$A_i^s Y_i^s = \widetilde{U_{i-1}^s}, \quad Y_i^s = \begin{pmatrix} G_i^s \\ \vdots \\ H_i^s \end{pmatrix}, \quad \begin{matrix} s = 0 \ldots k-1, \\ i = 3, 5 \ldots p/2^s - 1. \end{matrix} \tag{6}$$

We further let $D = A$, with similar notations for D_i^s as in (2),

$$D_i^s, \quad i = 1, 2, \ldots, p/2^s, \quad s = k, k-1, \ldots, 0. \tag{7}$$

We describe the main stages of the algorithm in the following:

For $s = k, k-1, \ldots, 0$ do in parallel

$$D_i^s = A_i^s - U_{i-1}^s (D_{i-1}^s)^{-1} L_{i-1}^s, \quad i = 2j,\ j = 1, 2, \ldots, p/2^{s+1}. \tag{8}$$

Find in parallel the Cholesky decompositions,

$$D_i^0 = \mathcal{L}_i \mathcal{L}_i^t, \quad i = 1, 2, \ldots, p. \tag{9}$$

Let,

$$\mathcal{L}_i = \begin{pmatrix} L_{i,1} & & & & \\ U_{i,1} & L_{i,2} & & & \\ & \ddots & \ddots & & \\ & & U_{i,l-2} & L_{i,l-1} & \\ & & & U_{i,l-1} & L_{i,l} \end{pmatrix}. \tag{10}$$

Solve in parallel the triangular systems

$$L_{i,l} L_{i,x} = L_{il}, \quad i = 1, 2, \ldots, p-1. \tag{11}$$

The decomposition $A = LL^t$ is then given by

$$L = \begin{pmatrix} \mathcal{L}_1 & & & & \\ L_{1,x}^t & \mathcal{L}_2 & & & \\ & \ddots & \ddots & & \\ & & L_{p-2,x}^t & \mathcal{L}_{p-1} & \\ & & & L_{p-1,x}^t & \mathcal{L}_p \end{pmatrix}. \tag{12}$$

THEOREM 2.1. *Let $\aleph_i \in M(m), i = 0,1,\ldots,p-1$ be given by*

$$(D_i^s)^{-1}\widetilde{L}_i^s = \begin{pmatrix} \vdots \\ \aleph_{i2^s} \end{pmatrix}, \quad i = 2j+1,\ j = 0,1,\ldots,p/2^{s+1}-1, \tag{13}$$

for $s = 0,1,\ldots,k$. Then,

$$\aleph_{i2^s} = F_i^s + H_i^s\aleph_{j2^s+1}(I - G_i^s\aleph_{j2^s+1})^{-1}E_i^s, \quad \aleph_0 \equiv 0. \tag{14}$$

Proof. see[2]. □

COROLLARY 2.2. *Suppose that the submatrices*

$$\begin{array}{ll} E_i^s, F_i^s & i = 1,3,\ldots,p/2^s-1 \\ G_i^s, H_i^s & i = 3,\ldots,p/2^s-1 \end{array} \quad s = 0,1,\ldots,k, \tag{15}$$

arc '-own. Then we can implement each step of the first stage of the algorithm in $O(m^3)$ operations.

3. Computing the E's,F's,G's,H's.. The algorithm we present is part of the author parallel algorithm for solving band s.p.d. systems of linear equations, see[1] section 2.1, and we will review here its main new ideas. We denote the *ith* row of $A^s, s = 0,1,\ldots,k$, given in (2), by

$$\mathcal{T}_i^s = \begin{pmatrix} 0 & \widetilde{U}_{i-1}^s & A_i^s & \widetilde{L}_i^s & 0 \end{pmatrix} \qquad i = 1,2,\ldots,p/2^s (16)$$

Step 0. We diagonalize the submatrices $A_i^0, i = 1,2,\ldots,p$. Let,

$$\mathcal{T}_i^0 = \left(0 \begin{array}{ccccc} \mathcal{G}_i^0 & \mathcal{V}_i^0 & 0 & 0 & \mathcal{E}_i^0 \\ \vdots & & \ddots & & \vdots \\ \mathcal{H}_i^0 & 0 & 0 & \mathcal{W}_i^0 & \mathcal{F}_i^0 \end{array} 0 \right), \tag{17}$$

debote the corresponding rows of A^0 after step 0.

Step s=1,2,...,k. The *ith* row of A_i^s has now the form

$$\mathcal{T}_i^s = \begin{pmatrix} \mathcal{T}_{2i-1}^{s-1} \\ \mathcal{T}_{2i}^{s-1} \end{pmatrix}, \qquad i = 1,2,\ldots,p/2^s,$$

$$\mathcal{T}_i^s = \left(\begin{array}{cccccccc} \mathcal{G}_{2i-1}^{s-1} & \mathcal{V}_{2i-1}^{s-1} & 0 & 0 & \mathcal{E}_{2i-1}^{s-1} & & & \\ & & * & & & & 0 & \\ \mathcal{H}_{2i-1}^{s-1} & 0 & 0 & \mathcal{W}_{2i-1}^{s-1} & \mathcal{F}_{2i-1}^{s-1} & & & \\ & & & \mathcal{G}_{2i}^{s-1} & \mathcal{V}_{2i}^{s-1} & 0 & 0 & \mathcal{E}_{2i}^{s-1} \\ & 0 & & & & * & & \\ & & & \mathcal{H}_{2i}^{s-1} & 0 & 0 & \mathcal{W}_{2i}^{s-1} & \mathcal{F}_{2i}^{s-1} \end{array}\right),$$

with $\mathcal{V}_j^s = \mathcal{V}_{(j-1)2^s+1}^0$, and $\mathcal{W}_j^s = \mathcal{W}_{j2^s}^0$. We eliminate $\mathcal{G}_{2i}^{s-1}, \mathcal{F}_{2i-1}^{s-1}$ in parallel by subtracting $\mathcal{G}_{2i}^{s-1}(\mathcal{W}_{2i-1}^{s-1})^{-1}$ times the last row of $\mathcal{T}_{2i-1}^{s-1}$ from the first row of $\mathcal{T}_{2i}^{s-1}$, and $\mathcal{F}_{2i-1}^{s-1}(\mathcal{V}_{2i}^{s-1})^{-1}$ times the first row of $\mathcal{T}_{2i}^{s-1}$ from the last row of $\mathcal{T}_{2i-1}^{s-1}$. As a result we obtain

$$\mathcal{T}_i^s = \left(\begin{array}{cccccccc} \mathcal{G}_{2i-1}^{s-1} & \mathcal{V}_{2i-1}^{s-1} & 0 & 0 & \mathcal{E}_{2i-1}^{s-1} & & 0 & \\ & & * & & & & & \\ \mathcal{H}_{2i-1}^{s-1} & 0 & 0 & \mathcal{R}_{2i-1}^{s-1} & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & \mathcal{S}_{2i}^{s-1} & 0 & 0 & \mathcal{E}_{2i}^{s-1} \\ & & & & & * & & \\ & 0 & & \mathcal{H}_{2i}^{s-1} & 0 & 0 & \mathcal{W}_{2i}^{s-1} & \mathcal{F}_{2i}^{s-1} \end{array}\right).$$

We then eliminate $\mathcal{H}_{2i}^{s-1}, \mathcal{E}_{2i-1}^{s-1}$ in parallel in a similar way, i.e., we subtract $\mathcal{H}_{2i}^{s-1}(\mathcal{R}_{2i-1}^{s-1})^{-1}$ times the last row of $\mathcal{T}_{2i-1}^{s-1}$ from the last row of $\mathcal{T}_{2i}^{s-1}$, and $\mathcal{E}_{2i-1}^{s-1}(\mathcal{S}_{2i}^{s-1})^{-1}$ times the first row of $\mathcal{T}_{2i}^{s-1}$ from the first row of $\mathcal{T}_{2i-1}^{s-1}$. The *ith* row of A^s finally receives a similar form to (17), i.e.,

$$\mathcal{T}_i^s = \begin{pmatrix} \mathcal{G}_i^s & \mathcal{V}_i^s & 0 & 0 & \mathcal{E}_i^s \\ & & * & & \\ \mathcal{H}_i^s & 0 & 0 & \mathcal{W}_i^s & \mathcal{F}_i^s \end{pmatrix}. \tag{18}$$

THEOREM 3.1. *Let $\mathcal{T}_i^s, s = 0,1,\ldots,k$ be as in (16). Then at the end of step s of the algorithm*

$$E_i^s = (\mathcal{V}_i^s)^{-1}\mathcal{E}_i^s,\ F_i^s = (\mathcal{W}_i^s)^{-1}\mathcal{F}_i^s, \quad \begin{array}{l} s = 0,1,\ldots,k, \\ i = 1,3,\ldots,p/2^s-1 \end{array}, \tag{19}$$

and

$$G_i^s = (\mathcal{V}_i^s)^{-1}\mathcal{G}_i^s,\ H_i^s = (\mathcal{W}_i^s)^{-1}\mathcal{H}_i^s, \quad \begin{array}{l} s = 0,1,\ldots,k-1, \\ i = 3,5,\ldots,p/2^s-1 \end{array}. \tag{20}$$

Proof. The first *s* steps of the algorithm may be viewed as an elimination procedure applied to the linear systems

$$A_i^s X_i^s = \widetilde{L}_i^s, \qquad A_i^s Y_i^s = \widetilde{U}_{i-1}^s, \qquad i = 1,2,\ldots,p/2^s, \tag{21}$$

in (5) and (6). The result now follows easily from (18). □

Complexity and Processor assignments. In *Step* 0, we assign processor number $i = 1,2,\ldots,p$, to the *ith* row of A^0. Each processor then diagonalize its corresponding submatrix in parallel in approximately $2nm^2/p$ operations. In *Step* $s = 1,2,\ldots,k$ the pair of processors

$$(i-1)2^s+1, i2^s \qquad i = 1,2,\ldots,p/2^s, \tag{22}$$

is assigned to rows $2i-1, 2i$ of A^{s-1}. Each such pair, performs the corresponding elimination step, exchanging $O(m^2)$ information and performing $O(m^3)$ operations. Hence, for $m,p \ll n$, the algorithm performs approximately

$$2nm^2/p + O(m^3 log\ p) \sim 2nm^2/p \ \text{ operations.} \tag{23}$$

4. A pseudo code. Let A be a band s.p.d matrix of order $n = qm$ and bandwidth m, and let $A = LL^t$ be the Cholesky decomposition of A as in Section 2. We assume for simplicity that $q = lp$ where $p = 2^{k+1}$ is the number of processors.

Bottom-up sweep: We compute the E's,F's,G's,H's. *Step s=0.* We compute in parallel:

$$\begin{array}{lll} E_i^0 = F_i^0 = A_i^{-1}L_i, & \text{for} & i = 1,3,\ldots,p-1. \\ G_i^0 = H_i^0 = A_i^{-1}U_{i-1}, & \text{for} & i = 3,5,\ldots,p-1. \end{array} \tag{24}$$

Step s=1,2,...,k-1. We perform in parallel step *s* of the elimination procedure as in Section 3, and then compute in parallel:

$$\begin{array}{ll} E_i^s = (\mathcal{V}_i^s)^{-1}\mathcal{E}_i^s, \quad F_i^s = (\mathcal{W}_i^s)^{-1}\mathcal{F}_i^s, & i = 1,3,\ldots,p/2^s-1. \\ G_i^s = (\mathcal{V}_i^s)^{-1}\mathcal{G}_i^s, \quad H_i^s = (\mathcal{W}_i^s)^{-1}\mathcal{H}_i^s, & i = 3,5,\ldots,p/2^s-1. \end{array} \tag{25}$$

Step s=k. We perform step k of the elimination procedure and compute

$$F_1^k = (\mathcal{W}_1^k)^{-1}\mathcal{F}_1^k. \qquad (26)$$

Top-down sweep: We find $A = LL^t$.
Step s=k,k-1,...,0. We compute in parallel

$$\aleph_{i2^s} = F_i^s + H_i^s\aleph_{j2^{s+1}}(I - G_i^s\aleph_{j2^{s+1}})^{-1}E_i^s, \qquad \aleph_0 \equiv 0, \quad (27)$$

for $i = 2j+1,\ j = 0,1,\ldots,p/2^{s+1}-1$.

Step s=-1. We compute in parallel

$$A_{i+1} = A_{i+1} - U_i\aleph_i, \quad i = 1,2,\ldots,p-1. \qquad (28)$$

Find in parallel the decompositions

$$A_i = \mathcal{L}_i\mathcal{L}_i^t, \quad \text{for} \quad i = 1,2,\ldots,p, \qquad (29)$$

and finally, solve in parallel the linear systems

$$L_{i,i}L_{i,x} = L_{il} \quad \text{for} \quad i = 1,2,\ldots,p-1. \qquad (30)$$

The triangular factor L of $A = LL^t$ is then given as in (12).

Complexity and Processor assignments. We have considered the bottom-up sweep in Section 3. In *Step s=k,k-1,...,0* of the top-down sweep, processor numbers

$$i2^s, \quad i = 2j+1,\ j = 0,1,\ldots,p/2^{s+1}-1 \qquad (31)$$

compute the corresponding $\aleph_{i2^s}$, performing $O(m^3)$ operations and exchanging $O(m^2)$ information in parallel. Then, in *Step s=-1*, processor $i = 1,2,\ldots,p$, performs approximately nm^2/p operations. Hence, for $m,p \ll n$, the complexity is

$$O(m^3 log\ p) + nm^2/p \sim nm^2/p \text{ operations}, \qquad (32)$$

for a total of $\sim 3nm^2/p$ operations. We conclude that the algorithm can be efficiently implemented on many existing parallel computers with $S_p \sim p/3$ speed-up, and with low communication overhead.

5. An O(logm logn) time algorithm. Let A be a band s.p.d. matrix of order $n = qm$ and bandwidth m, and let $A = LL^t$ be the Cholesky decomposition of A. We show in this section how to implement the algorithm given $p > q/log\ q$ processors. The algorithm proceeds as for the case where there are $min(q,p)$ processors, only now matrix operations of order m are done in parallel. We show how to implement efficiently each matrix operation such as add, multiply, invert, and find the Cholesky decomposition, in parallel. We distinguish between two cases as follows:

$q/log\ q < p \le qm^2/log\ q$: We perform each step with as many processors as available. For example, with q processors, we implement each operation in *Step* 0 with one processor, and *Step* $s = 1,2,\ldots,k$, with 2^s processors. Using standard parallel methods, each of the above matrix operations require at least $O(m)$ time. As there are only $\sim 2log\ q$ steps the total complexity is

$$O(nm^2/p) + O(mlog\ q) = O(nm^2/p) \text{ time}, \qquad (33)$$

and $S_p \sim p/3$, as the total operation count remains the same.

$qm^2/log\ q < p \le q\mathcal{M}(m)/(log\ m\ log\ n)$. We proceeds as in the previous case, using only the basic matrix operations such as add, subtract, and multiply. Here, $\mathcal{M}(m) = m^\beta, 2 \le \beta \le 3$, and $m^\beta/log\ m$ denotes the least number of processors required in order to multiply matrices of order m in $O(log\ m)$ time. For example, using the standard multiplication algorithm $\beta = 3$, parallel implementation of Strassen's method gives $\beta = \log 7 \sim 2.8$ see Chandra[5], and for huge size matrices β can be reduced even further, see Pan and Reif[8].

The bottom-up sweep is implemented as in[1] section 3, with complexity $O(q\mathcal{M}(m)/p)$. In the top-down sweep we have the following:

1. We multiply matrices of order m in

$$O(\mathcal{M}(m)/p) \quad \text{time, using} \quad p \le \mathcal{M}(m)/log\ m \quad \text{processors.}$$

2. Consider the inverses

$$(I - G\aleph)^{-1} = (I - Z)^{-1}, \qquad Z = G\aleph,$$

in (27), where we denote for simplicity G_i^s by G, and $\aleph_{j2^{s+1}}$ by $\aleph$. Here, we observe from[2] that

$$\| (I - Z)^{-1} - S_M \| \le \sum_{i=M}^{\infty} \| Z \|^i \le \alpha^M/(1-\alpha), \quad \alpha < 1,$$

where,

$$S_M = \sum_{i=0}^{M-1} Z^i = \prod_{i=0}^{logM/2} (I + Z^{2^i}),$$

and the solution converges quadratically. For example, let $\alpha = 2^{-t/2^{10}}$ and let the precision used be $\epsilon = 2^{-t}$. Then for $t \ge 8$, eleven products suffices to get an accurate inverse. The actual convergence rate is clearly α dependent, and even for $\alpha = 1 - 1/m^{O(1)}$ the series converges in $O(log\ m)$ steps. We will assume for simplicity that α is independent of m. The actual properties of this new iterative method requires further research.

3 We finally find the Cholesky decomposition of the dense s.p.d matrices A_i^0 in (29), using the author new divide and conquer parallel algorithm described in[3]. A similar iterative scheme used there gives a complexity of order

$$O(\mathcal{M}(m)/p) \quad \text{time using} \quad p \le \mathcal{M}(m)/log^2 m \quad \text{processors}.$$

As a by product we obtain the inverses of the triangular factors and the triangular systems in (30) can be solved by multiplication.

We conclude that the total complexity is

$$O(q\mathcal{M}(m)/p) \quad using \quad p \le q\mathcal{M}(m)/(log\ m\ log\ n) \quad processors,$$

with $S_p = O(p)$ speed-up over sequential algorithms. Furthermore, there are only $O(log\ q + log\ m) = O(log\ n)$ steps each dominated by the time to multiply matrices of order m, an operation which can be efficiently implemented on many parallel computers.

6. An application to the QR algorithm. The QR algorithm finds the eigenvalues of A by repeatedly performing

- Find the QR factorization $A = QR$.
- Set $A = RQ$.

until the offdiagonal elements become negligible. The resulting diagonal matrix is similar to the original matrix and therefore has the same eigenvalues. Sameh and Kuck have presented a parallel QR algorithm for tridiagonal matrices see[10] which can not be applied to general band matrices. We present for the first time such an efficient and fast parallel algorithm.

Let A be a band symmetric non-singular matrix of order $n = qm$ and bandwidth m, and let $A = QR$ be the QR decomposition of A. Then it is well known, see Parlett[9], that Q is lower Hessenberg of bandwidth m, and R is upper triangular of bandwidth $2m$, i.e.,

$$q_{ij} = 0,\ j < i - m, \quad \text{and} \quad r_{ij} = 0,\ j < i,\ j > i + 2m. \qquad (34)$$

Hence, RQ is lower Hessenberg of bandwidth m, but since $RQ = Q^tAQ$ is symmetric, it is again a band matrix of band width m. We have therefore the following algorithm:

- Set $B = A^tA = A^2$, a band s.p.d. matrix.
- Find the Cholesky decomposition $B = LL^t$.
- Solve the triangular system $LX = AL$.

Now, since $A = QR$ it follows that $L = R^t$ and therefore

$$X = L^{-1}AL = R^{-t}AR^t = Q^tR^t = RQ, \qquad (35)$$

is the transformed matrix sought. Let us denote X by

$$X = \begin{pmatrix} X_1 & Y_1 & & & \\ Z_1 & X_2 & Y_2 & & \\ & \ddots & \ddots & \ddots & \\ & & Z_{q-2} & X_{q-1} & Y_{q-1} \\ & & & Z_{q-1} & X_q \end{pmatrix}, \quad \begin{matrix} Z_i = Y_i^t \text{ is} \\ \text{upper triangular ,} \end{matrix}$$

and let

$$R = \begin{pmatrix} R_1 & S_1 & * & & \\ & R_2 & S_2 & * & \\ & & \ddots & \ddots & \ddots \\ & & & R_{q-1} & Sq-1 \\ & & & & R_q \end{pmatrix}$$

where all submatrices are of order m. Then,

$$AL = \begin{pmatrix} E_1 & F_1 & & & & \\ * & E_2 & F_2 & & & \\ * & * & E_3 & F_3 & & \\ * & * & * & E_4 & F_4 & \\ \ddots & \ddots & \ddots & \ddots & \ddots & \\ & * & * & * & E_{q-1} & F_{q-1} \\ & & * & * & * & E_q \end{pmatrix}, \quad \begin{matrix} E_i, F_i \in M(m), \\ \\ F_i \text{ is lower} \\ \text{triangular ,} \end{matrix}$$

and therefore X can be found as follows:

Solve in parallel the triangular systems

$$R_i^tY_i = F_i, \qquad \text{for } i = 1,2,\ldots,q-1. \qquad (36)$$

Solve in parallel the triangular systems

$$R_i^tX_i = E_i - S_{i-1}^tY_{i-1}, \qquad \text{for } i = 1,2,\ldots,q, \qquad (37)$$

Complexity: The complexity of the algorithm is dominated by the time to find the decomposition $B = R^tR$.

7. Conclusion.. We have presented a practical parallel algorithm for finding the Cholesky decomposition of a general band s.p.d. matrix that does not implement the known sequential method. We have suggested a divide and conquer approach whose inherent tree structure make it simple to implement on many parallel computers. Moreover, in most practical cases the operation count is approximately only as thrice as the sequential method and the overhead due to interprocessor communication is negligible.

Other numerical algorithms such as the conjugate gradient algorithm for solving sparse systems of linear equations, and Lanczos methods for finding the eigenvalues of symmetric sparse matrices are also sequential in nature and we believe that more parallel oriented algorithms for these problems can be found.

REFERENCES

[1] I. Bar-On, *A practical parallel algorithm for solving band symmetric positive definite systems of linear equations*, ACM Trans. Math. Softw., 13 (1987), pp. 323–332.

[2] ——, *Efficient logarithmic time parallel algorithms for the Cholesky decomposition of band matrices*, Tech. Report 661, Technion, Computer Science Department, 1990.

[3] ——, *New divide and conquer parallel algorithms for the Cholesky decomposition and Gram-Schmidt process*, Tech. Report 665, Technion, Computer Science Department, 1990.

[4] R. P. Brent and F. T. Luk, *Computing the Cholesky factorization using a systolic architecture*, in Proc. Sixth Australian Computer Science Conf., 1982, pp. 295–302.

[5] A. K. Chandra, *Maximal parallelism in matrix multiplication*, Tech. Report RC-6193, I.B.M. Watson Research Center, Yorktown Heights, N.Y., 1976.

[6] J. J. Dongarra, A. Sameh, and D. Sorensen, *Implementation of some concurrent algorithms for matrix factorization*, Parallel Computing, 3 (1986), pp. 25–34.

[7] S. P. Kumar and J. S. Kowalik, *Parallel factorization of a positive definite matrix on an MIMD computer*, in International Conference on Parallel Processing, 1984, pp. 410–416

[8] V. Pan and J. H. Reif, *Efficient parallel solution of linear systems*, in Proc. Seventeenth Annual Symposium on the Theory of Computing, 1985, pp. 143–152.

[9] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs., 1980.

[10] A. H. Sameh and D. J. Kuck, *A parallel QR algorithm for symmetric tridiagonal matrices*, IEEE Trans. Comput., C-26 (1977), pp. 147–152.

A RECURSIVE DOUBLING ALGORITHM FOR SOLUTION OF SOME SECOND ORDER RECURRENCES ON HYPERCUBE MULTIPROCESSORS

AYŞE KİPER
Department of Computer Engineering,
Middle East Technical University,
06531, Ankara-Turkey

Abstract. The second-order linear recurrence formulae which results from the Fourier series coefficients of the Jacobian elliptic functions $sn^m(u,k), cn^m(u,k)$, and $dn^m(u,k)$ with $m \geq 1$, are evaluated by the method of recursive doubling on an Intel iPSC/d4 and performance results are discussed.

1. INTRODUCTION.

In the evaluation of functions by series approximations, the accuracy increases with the increase in the number of terms of expansions which results in an increase in computer time.

The Fourier series expansion for the twelve Jacobian elliptic functions (JEFs) have been studied and given by several authors (Abramowitz and Stegun [1], Byrd and Friedman [2], Du Val [3] Whittaker and Watson [10]. Two-term recurrence formulae have been obtained for the coefficients of these series corresponding to powers of the JEFs (Kiper [5]). The resulting recurrence formulae are of the second order and linear. Parallel evaluation of these recurrences using nested recurrent product form algorithm and using method of recursive doubling on a MIMD system was considered by Kiper [6] and Kiper and Evans [7], respectively. In this paper an implementation of the recursive doubling algorithm on an Intel iPSC/d4 hypercube multiprocessor was developed for the evaluation of the mentioned recurrences and the results were discussed in terms of the speed-up and the efficiency of the parallel algorithm.

2. GENERALISATION OF RECURRENCE FORMULAE OF THE FOURIER COEFFICIENTS FOR POWERS OF THE JEFs.

The analysis of the relations for the Fourier coefficients for powers of the JEFs (Kiper [5]) shows that for a prescribed k (k is the modulus of the elliptic functions) these recurrences may be represented by the common expression

$$\left.\begin{array}{lcl} \omega_n^{(0)} & = & \alpha \\ \omega_n^{(1)} & = & \beta \\ \omega_n^{(r)} & = & a(n,r)\omega_n^{(r-1)} + b(r)\omega_n^{(r-2)}, \ r = 2,3,\ldots\ell \end{array}\right\} n = 0,1,\ldots m \tag{2.1}$$

where ℓ is the required power and m is the required number of terms in the expansion. Equation (2.1) is a second order linear recurrence relation $R(\ell,2)$.

3. PREVIOUS EVALUATIONS.

A sequential evaluation of the coefficients for the Fourier expansion of the JEFs $sn^m(u,k), cn^m(u,k)$ and $dn^m(u,k)$ with $m \geq 1$ were obtained and the numerical values are given for various values of k ($0.1 \leq k^2 \leq 0.9$) (Kiper [5]). It is seen that the rate of convergence decreases as the value of k and the power of the JEFs increase.

A parallel evaluation of the coefficients was formulated by Kiper [6] using the nested recurrent product form algorithm in which the relation (2.1) has been expressed as the solution of a matrix system

$$\underline{\omega} = A\underline{\omega} + \underline{b} \tag{3.1}$$

where

$$A = \begin{bmatrix} 0 & & & & & \\ a(n,3) & 0 & & & & \\ b(4) & a(n,4) & 0 & & & \\ 0 & & & & & \\ & & & & & \\ & & & & & \\ 0 & & & b(\ell) & a(n,\ell) & 0 \end{bmatrix}$$

$$\underline{\omega} = \begin{bmatrix} \omega_n^{(2)} \\ \omega_n^{(3)} \\ \\ \\ \\ \omega_n^{(\ell)} \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} a(n,0)\,\omega_n^{(1)} + b(0)\,\omega_n^{(0)} \\ b(1)\,\omega_n^{(0)} \\ 0 \\ \\ \\ 0 \end{bmatrix}$$

If the first m terms of the expansion for a power ℓ is required, then a matrix system of size $(\ell - 1)$ must be solved $(m+1)$ times with $n = 0,1,2,\ldots m$. The size of the system increases with the increasing required power and the number of solutions of the system increases with the increasing number of terms.

Another parallel approach to the solution of the recurrence relation (2.1) was considered by Kiper and Evans [7]. Equation (2.1) yields a form of second order linear recurrent equation which generates a vector result in two dimensions. Since, if we let

$$V_n^{(r)} = \begin{bmatrix} \omega_n^{(r)} \\ \omega_n^{(r-1)} \end{bmatrix} \text{ and } A_n^{(r)} = \begin{bmatrix} a(n,r) & b(r) \\ 1 & 0 \end{bmatrix} \tag{3.2}$$

$r = 2,3,\ldots\ell$; $n = 0,1,2,\ldots m$, then (2.1) for the prescribed power ℓ can be written as (Modi [8], Schendel [9])

$$V_n^{(\ell)} = A_n^{(\ell)} \cdot A_n^{(\ell-1)} \cdot \ldots \cdot A_n^{(2)} \cdot V_n^{(1)}, \quad n = 0,1,2,\ldots m. \tag{3.3}$$

The associative property of matrix-matrix multiplication leads us to use the recursive doubling process [9] in $O(\log_2 \ell)$ steps for each n ($n = 0,1,2,\ldots m$). It must be noted that computations need $O(\ell)$ steps for each n in the sequential mode. The numerical experiments of the proposed algorithm were carried out on the Sequent Balance 8000 multiprocessor with 5 processors and a comparative discussion of the results were given in [7].

4. RECURSIVE DOUBLING ALGORITHM ON HYPERCUBE MULTIPROCESSORS.

An implementation of the recursive doubling algorithm whose formulation has been already given in [7] was considered on an Intel iPSC/d4 hypercube multiprocessor.

The algorithm consists of the following there steps:

```
Step 1: Compute $V[r]$ locally
        $V[r] = A[r] * V[r-1], \quad r = 2,3,\ldots,\ell$

          $p$:  the number of nodes
          $n$:  the problem size
          $\ell$:  $n/p$

Step 2: Use the recursive doubling to send and receive $V[\ell]$
        for $k = 1$ to $\log(p)$ do
        begin

        if      (MyNode $<= (p - 1 - 2^{k-1})$) then
                send $V[\ell]$ to (MyNode $+2^{k-1}$);
        if      (MyNode $>= 2^{k-1}$) then
        begin
                receive $tmp\_V\ell$;
                $V[\ell] = V[\ell] * tmp\_V\ell$;
        end;
       end;

Step 3: Compute $\omega[r]$
        if    (MyNode $<= p - 2$) then
              send $V[\ell]$ to (MyNode $+1$);
        if    (MyNode $< > 0$) then
              receive $tmp\_V\ell$;
        for   $k = 1$ to $\ell - 1$ do

     $V[k][0][0] = V[k][0][0]*tmp\_V\ell[0][0]+V[k][0][1]*tmp\_V\ell[1][0]$;
```

The algorithm can be analysed as:

Step 1: takes $4(\ell - 1)$ multiplication and $2(\ell - 1)$ addition steps.

Step 2: takes $8*\log(p)$ multiplication $4*\log(p)$ addition and $2*\log(p-1)$ communication steps.

Step 3: takes $2(\ell - 1)$ multiplication, $m - 1$ addition and $2p - 3$ communication steps.

The total number of arithmetic and communication steps involved are $(9\ell - 4 + 12 * \log(p))$ and $(2p - 3 + 2 * \log(p - 1))$ successively.

The major advantage of the hypercube multiprocessor implementation is that the iPSC/d4 is a circuit-switched machine (contrasts to the store-and-forward counterpart) and the neighboring nodes of each node have not been found out (in order to implement the recursive doubling algorithm) as Eğecioğlu, Koç and Laub [4] did in their paper.

5. EXPERIMENTAL RESULTS AND CONCLUSIONS.

The effect of multiple processes was investigated by running the program on 1, 2, 3, 8 and 16 nodes (p) successively and computing the total amount of time T_p (milliseconds) needed. The performance of the algorithm was measured in terms of the speed-up (Sp) and the efficiency (Ep). The numerical results with respect to the problem size and the number of nodes used are given in Table 1.

Table 1

p	1	2			4			8			16		
ℓ	Tl	Tp	Sp	Ep	Tp	Sp	Ep	Tp	Sp	Ep	Tp	Sp	Ep
16	1.0	1.6	0.60	0.30	3.5	0.28	0.07	9.3	0.10	0.01	21.0	0.05	0.00
32	1.2	2.1	0.60	0.30	3.9	0.32	0.08	9.9	0.13	0.02	21.2	0.06	0.00
64	2.5	2.9	0.85	0.42	4.3	0.57	0.14	9.9	0.25	C.03	21.2	0.12	0.01
128	4.9	4.8	1.03	0.51	5.2	0.94	0.24	10.2	0.48	0.06	21.5	0.23	0.01
256	9.9	8.4	1.17	0.59	7.1	1.40	0.35	11.3	0.88	0.11	22.1	0.45	0.03
512	19.8	15.9	1.24	0.62	10.8	1.83	0.46	13.0	1.52	0.19	23.3	0.85	0.05
1024	39.6	31.0	1.28	0.64	18.8	2.10	0.53	16.7	2.38	0.30	24.5	1.62	0.10
2048	99.6	63.7	1.56	0.78	34.2	2.91	0.73	25.4	3.93	0.49	28.6	3.49	0.22
4096	223.1	153.8	1.45	0.73	68.1	3.27	0.82	39.6	5.64	0.70	38.4	5.81	0.36
8192	467.1	333.0	1.40	0.70	158.6	95.0	0.74	78.1	5.98	0.75	56.2	8.30	0.52

Variations of the speed-up and the efficiency with the problem size are also given in graphical forms in Fig. 1 and Fig. 2 respectively.

As it will be seen from the numerical results, both the speed-up and efficiency improve with the increasing problem size. Also it can be recognised that the maximum values attained for speed-up and efficiency are proportional to the number of nodes used.

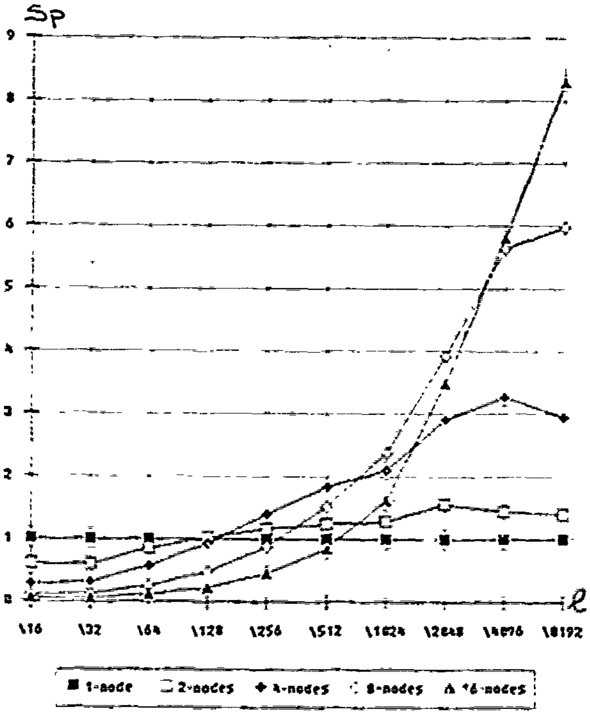


Fig. 1. Speed-up

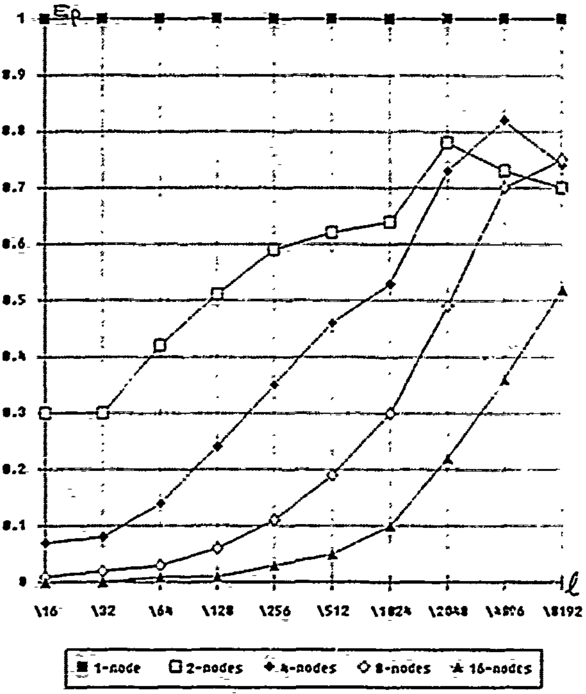


Fig. 2. Efficiency

Acknowledgment. The author acknowledges her thanks Mr. M.M. Lee at Michigan Technological University for carrying out the computations.

REFERENCES

[1] M. ABRAMOWITZ AND I.A. STEGUN, *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, Nat. Bur. standards, Appl. Math. Series No.55, December 1954 (Also: Dover, New York, 1968).

[2] P.F. BYRD AND M.D. FRIEDMAN, *Handbook of Elliptic Integrals for Engineers and Scientists*, 2nd ed., Springer-Verlag, Berlin 1971.

[3] P. DU VAL, *Elliptic Functions and Elliptic Curves*, London Mathematical Society Lecture Notes Series 9, Cambridge Univ. Press, Cambridge, 1973.

[4] Ö. EĞECİOĞLU, Ç.K. KOÇ AND J. LAUB, *A recursive doubling algorithm for solution of tridiagonal systems on hypercube multiprocessors*, J. Comp. Appl. Math., 27 (1989), pp.95-108.

[5] A. KİPER, *Fourier series coefficients for powers of the Jacobian elliptic functions*, Math. Comp. 43 (1984), pp.247-259.

[6] A. KİPER, *Some recurrence relations and their parallel evaluation using nested recurrent product form algorithm*, J. Comp. Appl. Math., 28 (1989), pp.231-235.

[7] A. KİPER AND D.J. EVANS, *Parallel evaluation of some recurrence relations by recursive doubling*, NATO/ASI Series on Supercomputing F.62, J.S. Kowalik, ed., Springer-Verlag, Berlin, 1989.

[8] J.J. MODI, *Parallel Algorithms and Matrix Computations*, Oxford University Press, New York, 1988.

[9] U. SCHENDEL, *Introduction to Numerical Methods for Parallel Computers*, Ellis Horwood, New York, 1984.

[10] E.T. WHITTAKER AND G.N. WATSON, *Modern Analysis*, Cambridge Univ. Press, Cambridge 1962.

MATRIX ALGEBRA AND HYPERCUBE PARALLEL TRANSMISSIONS

JAIME SEGUEL and JULIO BARETY
University of Puerto Rico at Mayaguez
Mayaguez, PR 00708.

Abstract. Using a matrix representation of a certain family of permutations we model hypercube parallel data transmissions. Our model provides a theoretical framework for the estimation of the number of parallel transmissions involved in a given algorithm as well as the algorithms that achieve the lowest number of parallel transmissions.

1.- Introduction. A d-dimensional hypercube architecture consists of 2^d node processors linked by the edges of a d-dimensional hypercube. The processors are numbered in a way such that the minimal number of edges between any pair of them is the number of different digits in the binary representation of their labels. Thus, two node-processors are joined if and only if their binary labels differ at exactly one bit. In such an architecture, a nonnumerical issue that is crucial to the performance of algorithms is the frequency and cost of communications among processors. These figures will depend on the underlying problem, on how the data is mapped onto the processors, and on the numerical algorithm. In this work we present a formal complexity analysis for the interprocessor communication problem associated to the product of a $2^d \times 2^d$ complex matrix times a 2^d-dimensional vector. Our analysis assumes that all interprocessor communications are describable by the action of a certain group of permutations on the nodes of the hypercube. A matrix representation of these permutations allows us to use matrix algebra techniques to estimate the minimal number of parallel transmissions involved in a given algorithm as well as to find the algorithms that achieve that estimated lower bound. In this paper we briefly present the theoretical foundations of our communication complexity model and use some techniques derived from it to analyse the complexity and design optimal communication algorithms for computing some members of a well known family of permutations.

2.- Theoretical Background. Given $x \in N_0 = \{0, 1, 2, \ldots\}$, we write $x = \sum_{n=0}^{\infty} c_n(x) 2^n$, where $c_n(x) \in \{0, 1\}$, for each n. We also set $S_x = \{n \in N_0 : c_n(x) \neq 0\}$. This finite set is called the *spectrum* of x. The largest integer in S_x is termed the *degree* of x. Given x and $y \in N_0$, one can define the so-called dyadic sum, $x \dot{+} y$, of x and y by $x \dot{+} y = \sum_n |c_n(x) - c_n(y)| 2^n$.

Proposition 1.- N_0, under dyadic addition, is an abelian group. In particular, 0 is the additive identity and every element in N_0 is its own inverse.

The initial segment of N_0, $G_{2^d} = \{0, 1, \ldots, 2^d - 1\}$ is a subgroup of N_0 under dyadic addition. Indeed, G_{2^d} is the direct product of d copies of $Z_2 = \{0, 1\}$, the abelian group of integers under addition modulo 2.
The *Hamming weight* $w(x)$, of $x \in N_0$, is defined by $w(x) = \sum_n c_n(x)$. The addition here is the ordinary addition. One can see that:

Proposition 2.- (a) $w(x) \geq 0$, $\forall x \in N_0$,
(b) $w(x \dot{+} y) \leq w(x) + w(y)$, $\forall x, y \in N_0$.

In particular, in an initial segment G_{2^d}, the Hamming weight ranges through the integers between 0 and d. Another measure of size used in this context is the Hamming distance. Given x and $y \in N_0$, the *Hamming distance*, $h(x, y)$, between x and y, is defined by $h(x, y) = \sum_n |c_n(x) - c_n(y)|$. Since $c_n(x \dot{+} y) = |c_n(x) - c_n(y)|$, it is clear that $h(x, y) = w(x \dot{+} y)$. Some basic properties of the Hamming distance are given in the following proposition

Proposition 3.- (a) For any $x, y, z \in N_0$, $h(x \dot{+} z, y \dot{+} z) = h(x, y)$
(b) If $S_z \bigcap (S_x \bigcup S_y) = \emptyset$, then $h(x, y \dot{+} z) = h(x \dot{+} z, y) = h(x, y) + h(z, 0)$
(c) If $d >$ max { degree w, degree z }, then, for any x, $y \in N_0$,

$$h(2^d x \dot{+} z, 2^d y \dot{+} w) = h(x, y) + h(z, w).$$

Note that if x and $y \in G_{2^d}$ are *complementary*, i.e. $c_n(x) \neq c_n(y)$, $\forall n$, then $h(x, y) = d$. This, in fact, is a necessary and sufficient condition for two integers in G_{2^d} to be complementary.
A permutation $\sigma : G_{2^d} \to G_{2^d}$ will be called a *nearest neighbor transmission* (NNT) permutation, if $h(x, \sigma(x)) = 1$, $\forall x \in G_{2^d}$. Thus, such permutations modify a single bit in the binary representation of each $x \in G_{2^d}$. The NNT permutation will be called *perfect* if the bit modified is always the same. For instance, $\sigma_1(x) = x \dot{+} 1$ and $\sigma_2(x) = x \dot{+} 2$ are perfect NNT permutations in G_{2^2}. In general, it is clear that for each $d = 1, 2, \ldots$, there are d perfect NNT permutations on G_{2^d} and they are of the form $\sigma_j(x) = x \dot{+} 2^j$, $\forall x \in G_{2^d}$, $j = 0, 1, \ldots, d - 1$. Let C be the set of all complex numbers, $M_{2^d}(C)$ the linear space of all $2^d \times 2^d$ complex matrices and $GL_{2^d}(C)$ the linear space of non-singular matrices with complex entries. In order to get a matrix representation for the NNT permutations we first define the matrix representation of Z_2, $R : Z_2 \to M_2$, as

$$R(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2 \text{ and } R(1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = J_2.$$

Let $\otimes$ denote the tensor (also Kronecker or direct) product of matrices. Since G_{2^d} is the direct product of d copies of Z_2, the map $K(\ , d) : x \in G_{2^d} \to K(x, d) \in GL_{2^d}(C)$, where

$$K(x, d) = \underbrace{R(c_{d-1}(x)) \otimes \ldots \otimes R(c_0(x))}_{d}$$

is a matrix representation of G_{2^d}.
The d perfect NNT permutations on G_{2^d} are represented by the matrices $K(2^j, d)$, $j = 0, 1, \ldots, d - 1$. In general, being $K(\ , d)$ a representation, $K(x \dot{+} y, d) = K(x, d) K(y, d)$ and the set $U_{2^d} = \{K(x, d) . x \in G_{2^d}\}$ is a commutative subgroup of $GL_{2^d}(C)$. Furthermore, $K(x, d)$ is its own inverse. If $d = 2$, the elements in U_{2^2} are

$$K(0, 2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, K(1, 2) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$K(2, 2) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \text{ and } K(3, 2) = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Proposition 4.- Let $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{2^d - 1})$ and let

$$D[\lambda] = \begin{pmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_{2^d - 1} \end{pmatrix}.$$

Then, for all $y \in G_{2^d}$, $K(y, d) D[\lambda] K(y, d) = D[(\lambda_{0 \dot{+} y}, \lambda_{1 \dot{+} y}, \ldots, \lambda_{(2^d - 1) \dot{+} y})]$.
Definition.- For any matrix $A \in M_{2^d}$ and $x \in G_{2^d}$ we define the $2^d \times 2^d$ diagonal matrix

$$D(x, A) = D[a_{0, x}, a_{1, 1 \dot{+} x}, \ldots, a_{2^d - 1, (2^d - 1) \dot{+} x}]$$

We have the following important result.
Theorem 1.- Any matrix $A \in M_{2^d}$ can be written as

$$A = \sum_{x \in G_{2^d}} D(x, A) K(x, d).$$

This representation is unique in the sense that if $A = \sum_{x \in G_{2^d}} B_x K(x, d)$, where each B_x is a diagonal matrix, then, necessarily $B_x = D(x, A)$, $\forall x \in G_{2^d}$.

Corollary 1.- For any matrix $A \in M_{2^d}$, the following decomposition holds

$$A = \sum_{x \in G_{2^d}} K(x, d) D(x, A^T),$$

where A^T is the tranposed matrix of A.

Example:

$$\begin{pmatrix} 2 & 1 & 3 & 4 \\ -7 & 3 & 0 & 8 \\ -3 & 5 & 1 & \sqrt{2} \\ \sqrt{7} & -1 & 0 & 7 \end{pmatrix} = \begin{pmatrix} 2 & & & \\ & 3 & & \\ & & 1 & \\ & & & 7 \end{pmatrix} I_4 + \begin{pmatrix} 1 & & & \\ & -7 & & \\ & & \sqrt{2} & \\ & & & 0 \end{pmatrix} I_2 \otimes J_2$$

$$\begin{pmatrix} 3 & & & \\ & 8 & & \\ & & -3 & \\ & & & -1 \end{pmatrix} J_2 \otimes I_2 + \begin{pmatrix} 4 & & & \\ & 0 & & \\ & & 5 & \\ & & & \sqrt{7} \end{pmatrix} J_2 \otimes J_2.$$

By using the matrix representation given in theorem 1 and its uniqueness, we can easily compute the generalized diagonals of ordinary and tensor product of matrices. In fact,

Theorem 2.- (a) Let $A \in M_{2^{d_1}}$ and $B \in M_{2^{d_2}}$. Then for any $z \in G_{2^{d_1 + d_2}}$,

$$D(z, A \otimes B) = D(x, A) \otimes D(y, B)$$

where x and y are the unique elements in $G_{2^{d_1}}$ and $G_{2^{d_2}}$ respectively, such that $z = x 2^{d_2} \dot{+} y$.

(b) Let $A, B \in M_{2^d}$. Then, for any $z \in G_{2^d}$

$$D(z, AB) = \sum_{x,y \in G_{2^d} : x \dotplus y = z} D(x, A) D(y, B_x)$$

where $B_x = K(x,d)BK(x,d)$.

The following quantity will be useful in the estimation of the number of parallel transmissions involved in computing with A. Given any $A \in M_{2^d}(C)$, $A \neq [0]$, we define $\delta(A) = \max\{w(x) : D(x, A) \neq [0]\}$. We call this quantity the *Hamming diameter* of A.

Proposition 5.- (a) Given any $A \in M_{2^{d_1}}$, $B \in M_{2^{d_2}}$, $\delta(A \otimes B) = \delta(A) + \delta(B)$.
(b) Given any A, $B \in M_{2^d}$, then $\delta(AB) \leq \min\{d, \delta(A) + \delta(B)\}$ This inequality can be strict even if both matrices are different from the zero matrix.

3.- Hypercube Parallel Transmissions. Our hypercube machine consists of 2^d node processors. Each processor is endowed with a certain number of vector registers of length 2^t. Given a vector register v_1 in processor P_1 and a vector register v_2 in processor P_2, it will be always possible to send the content of the j-th component of v_1 to the j-th component of v_2. Furthermore, simultaneous exchanges of the contents of the j-th components of any two vector registers are allowed. Thus, if we assume that at a certain stage of a computational process, the 2^{d+t}-dimensional data vector c is loaded as a two-dimensional array: $[c(x,j)]$, $x \in G_{2^d}$ and $j \in G_{2^t}$, where x and j are the processor's and vector register component's labels respectively, all possible interprocessor data movements are representable by a set of permutations on G_{2^d}, $\{P_j : j \in G_{2^t}\}$. Each of these permutations acts on the segment $c_j = (c(0,j), c(1,j), ..., c(2^d - 1, j))$ of c, and therefore the whole action of the permutations P_j corresponds to the action of the direct sum $P = \oplus_{j \in G_{2^t}} P_j$ on c, antilexicographically ordered. The permutation P will be called an *hypercube parallel communication* (HPC). By representing each P_j in terms of our formulas in section 2,

$$P = \oplus_{j \in G_{2^t}} (\sum_{x \in G_{2^d}} D(x, P_j) K(x, d)).$$

An *hypercube parallel transmission* (HPT) is now defined as the special type of HPC where $P_j = D_j K(2^{r(j)}, d)$. Here D_j is a diagonal matrix with 1's and 0's on its diagonal and $0 \leq r(j) \leq d - 1$. If $r(j) = r$ for all j,

$$P = (\oplus_{j \in G_{2^t}}) (I_{2^t} \otimes K(2^r, d)).$$

Any HPC is performed through a sequence of HPT's. A lower bound for the number of HPT's involved in an hypercube parallel communication is $\max\{\delta(P_j) . j \in G_{2^t}\}$. This lower bound, however, is not always achievable

Theorem 3.- Any permutation on a 2^d-dimensional data vector is computable with a minimum number of HPTs.

Proof: Let A be a matrix representation of a permutation on C^{2^d}. Then $A = \sum_{x \in G_{2^d}} D(x, A) K(x, d)$. Let $Im\, D(x, A)$ and $Im\, D(x, A^T)$ be the image subspaces of $D(x, A)$ and $D(x, A^T)$ respectively. Since A is a permutation, $C^{2^d} = \oplus_{x \in G_{2^d}} Im\, D(x, A) = \oplus_{x \in G_{2^d}} Im\, D(x, A^T)$, and the permutation A can be written as the direct sum $A = \oplus_{x \in G_{2^d}} T_x$, where T_x is the restriction of $K(x, d)$ to $Im\, D(x, A^T)$. Now, the minimum number of HPTs required by A is equal to the $\max\{\delta(T_x) : x \in G_{2^d}\}$, which is, in fact, equal to $\delta(A)$.

Theorem 4.- Let $A \in M_{2^{d+t}}$ be such that $A = B \otimes C$, with $B \in M_{2^d}$ and $C \in M_{2^t}$. Let's also assume that computing with B requires $\delta(B)$ HPTs. Then computing with A also requires $\delta(B)$ HPTs.

Proof: Since $A = B \otimes C = (B \otimes I_{2^t})(I_{2^d} \otimes C)$, the process of computing $z = Au$ can be made in two steps, (1) $y = (I_{2^d} \otimes C)u$ and (2) $z = (B \otimes I_{2^t})y$. To implement Step (1) we divide u into 2^d segments u_x, $x \in G_{2^d}$, each of length 2^t. Then we load u_x in processor x and compute $y_x = Cu_x$. This computation requires no interprocessor communications. As for step (2), according to Proposition 6, (a), $\delta(B \otimes I_{2^t}) = \delta(B) + \delta(I_{2^t}) = \delta(B)$.

Theorem 4 establishes that a significant reduction in the number of HPTs involved in computing with A is achievable whenever A supports a tensor product factorization. This result explains, for instance, the good communication properties shown by the Cooley-Tuckey FFT.
Besides estimating the HPT complexity of any algorithm expressable in matrix language, our model helps in finding the best interprocessor data flow alternatives. We will consider, by way of example, algorithms computing certain permutations. First, we need to set up some corollaries of Proposition 5.

Corollary 1.- If $\sigma . G_{2^d} \rightarrow G_{2^d}$ is a permutation such that $\sigma^2 = id$ and if we write $D[\psi^x] = D[\psi_0^x, \psi_1^x, ..., \psi_{2^d-1}^x] = D(x, P_\sigma)$ where P_σ is a matrix representation of σ; then $K(y,d)\psi^x = K(x,d)[K(y,d)\psi^x]$.

Definition. Let $A \in M_{2^d}$. We define $\Sigma_A = \{x \in G_{2^d} : D(x, A) \neq [0]\}$.

Corollary 2.- If $\sigma : G_{2^d} \rightarrow G_{2^d}$, then (a) $\Sigma_A = \{j \dotplus \sigma^{-1}(j) : j \in G_{2^d}\}$,
(b) For any $j \in G_{2^d}$, there is one and only one $x \in \Sigma_A$ such that $\psi_j^x = 1$.

Now, we apply our theoretical framework to the analysis and design of algorithms for computing well known permutations such as the bit-reversal and the index digit permutations. For instance, let's consider $\beta_3 : G_{2^3} \rightarrow G_{2^3}$ the 3-digit bit-reversal permutation. Then β_3 is determined by the transpositions (1 4), (3 6). Now, for $i = 0, 1, 2, 3$; $2i \dotplus \beta_3(2i) = 0$ and $2i \dotplus 1 \dotplus \beta_3(2i + 1) = 5$. Thus, $D(0, P_{\beta_3}) = D[1,0,1,0,0,1,0,1]$ and $D(5, P_{\beta_3}) = D[0,1,0,1,1,0,1,0]$. Therefore, β_3 is reduced to $K(5,3)$ acting on the space $Im\, D(5, P_{\beta_3})$. Now, since $w(5) = 2$, $\delta(K(5,3)) = 2$ and therefore the HPT complexity of computing β_3 is 2. An algorithm achieving that complexity is obtained by simply factoring $K(5,3) = K(1,3)K(4,3)$.
Our second example starts with the permutation r determined by the cycle (0 1 2 3). Such a cycle induces a permutation $\sigma . G_{2^4} \rightarrow G_{2^4}$. This permutation is defined as

$$\sigma(c_3(x)2^3 + c_2(x)2^2 + c_1(x)2 + c_0(x)) = c_{r(3)}(x)2^3 + c_{r(2)}2^2 + c_{r(1)}2 + c_{r(0)}$$

and sometimes termed index-digit permutation. A direct calculation shows that $\Sigma_{P_\sigma} = \{0,3,5,6,9,10,12,15\}$. Since $\max\{w(x) : x \in \Sigma_{P_\sigma}\} = w(15) = 4$, $\delta(P_\sigma) = 4$. The algorithm performing P_σ in four HPTs is obtained by restricting the actions of $K(x,d)$ to $Im\; D(x, P_\sigma)$, for $x \in \Sigma_{P_\sigma}$. For x in Σ_{P_σ}, $x \neq 15$, the operator $K(x,d)$ factors as a product of two perfect NNT permutations. However, since $w(15) = 4$, $K(15,4)$ will factor as a product of four perfect NNTs.

References

[1] Swarztrauber, P., "Multiprocessor FFTs", Parallel Computing 5, pp. 197-210, 1987.

[2] Serre, J.P., "Linear Representations of Finite Groups", Springer-Verlag, 1977.

[3] Fraser, D., "Array permutation by index-digit permutation", J. ACM 22, pp. 189-191, 1981.

[4] Gannon, D. and Rosendale, J. "On the impact of communication complexity on the design of parallel numerical algorithms", IEEE Trans. Comput. 33, pp. 1180-1194, 1984.

Doping-induced anchoring transitions at liquid crystal surfaces

P. I. C. Teixeira and T. J. Sluckin

Faculty of Mathematical Studies, University of Southampton
Southampton SO9 5NH, United Kingdom

Abstract. We have generalised earlier work on anchoring of nematic liquid crystals by Sullivan and by Sluckin and Poniewierski in order to study transitions which may occur in binary mixtures of nematic liquid crystals as a function of concentration. Possible phase diagrams of anchoring angle versus dopant concentration have been calculated for a simple liquid crystal model.

1. INTRODUCTION

If a nematic liquid crystal (LC) is spread on top of an anisotropic substrate, the energy of the LC molecules at the nematic-substrate interface will depend on orientation, thus inducing preferential alignment of the nematic director[1,2,3].This phenomenon, known as *anchoring*, plays a crucial role in the fabrication of LC display devices, which rely on a delicate control of anchoring surfaces (see e.g. ref. 3). However, the underlying physical mechanisms are not yet fully understood.

Recently it was shown that discontinuous changes in anchoring direction - *anchoring transitions* - can occur as a result of changing concentrations of one or more adsorbates[4]. We are concerned with modelling the anchoring transition observed by Pieranski *et al*[5,6], which is driven by changes in the amount of water vapour adsorbed on a gypsum substrate. This is a transition between two monostable planar anchorings, which is notoriously difficult to study theoretically. We therefore started by considering a simpler case, that of a transition between homeotropic and planar anchorings.

2. THEORY

Following Sullivan and co-workers[7,8,9], we used a mean-field approximation to the Helmholtz free energy functional of a non-uniform nematic liquid to derive a simpler, Landau-de Gennes free energy functional. The main advantage of this method is that we obtain explicit expressions for the phenomenological coefficients appearing in Landau-de Gennes theory in terms of the intermolecular (and surface) potentials. Besides mean-field, the basic approximation is the assumption of a step-function variation of the density and order parameter profiles. Although this is clearly incorrect, as it neglects the role of surface adsorption[10,11], we expect the resulting theory to provide at least qualitative insight into the mechanism determining the equilibrium alignment.

We first generalised Sullivan and co-workers' approach to consider a single-component uniform nematic phase at a (plane) surface. Neglecting biaxiality, the surface term in the expression of the surface tension is[9].

$$f_s = w_0 + w_2 P_2(\cos\psi) + w_4 P_4(\cos\psi)$$

where $\psi = \cos^{-1}(\hat{n}.\hat{k})$ is the *tilt angle*, i.e. the angle between the normal to the surface, $\hat{k}$, and the nematic director, $\hat{n}$, and P_n is the nth-order Legendre polynomial. We can estimate the orientation favoured by the surface by minimising only f_s instead of the full surface tension[9,11].Equation (1) truncated after the $P_2(\cos\psi)$ term is known in the literature as the Rapini-Papoular form of the surface anchoring energy[12], which has been extensively used in both theoretical and experimental work.

This theory is straightforwardly generalised to a binary LC mixture at a surface. f_s is still given by (1) and the w_n are linear combinations of $V^{ij}_3(l_1l_2l)$, the third moments of the coefficients in the spherical harmonic expansion of the intermolecular potential between species i and j[13] for (l_1l_2l) = (000), (220), (202), (222) and (224), and V_{ext}, the integrated LC-surface interaction. In particular, w_4 is a function of $V(224)$ only. The coefficients in the linear combinations are products of the total density and of the concentrations and order parameters of the two components.

Minimisation of f_s with respect to ψ yields the phase diagram shown in fig. 1.

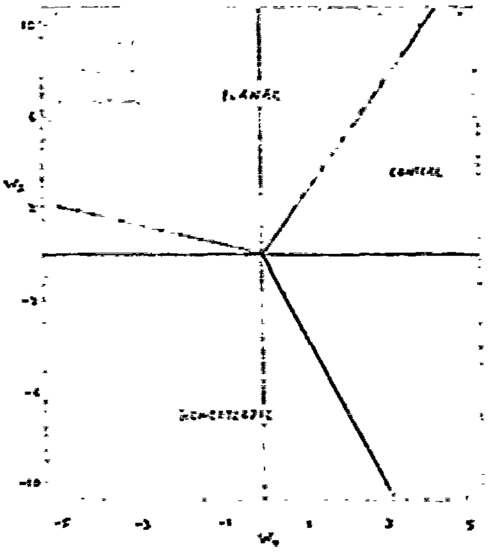


Figure 1: The anchoring phase diagram of a LC in (w_2,w_4) space.

As w_2 and w_4 change as functions of temperature and concentration, the trajectory of a system in (w_2,w_4) space may cross one or more of the boundaries between different anchoring domains, at which point an anchoring transition obtains which is first-order if $w_4 < 0$ or second-order if $w_4 > 0$. Sullivan and co-workers[7,9] argued that the $V(224;r)$ terms in the interaction potential f_s coming from quadrupole-quadrupole interactions and short-range anisotropic repulsive and attractive forces) are essentially positive, leading to non-negative w_4 and second-order transitions (however, w_4 may vanish, as we shall see).

In order to calculate w_n and ψ we need a specific model for $V(l_1 l_2 l;r)$. In a first approach, we used the simple Telo da Gama model of LCs[14], which includes (000), (220), (202) and (022) (=(202) terms, augmented with a (224) term coming from quadrupolar interactions[13] and an inverse-power law surface potential[10]. Note that, in this simple mean-field theory, only $(l_1 l_2 l)$ terms with l = 0 contribute to the free energy of the bulk phases[14] and therefore, given our step-function approximation for the density, concentration and order parameter profiles, the addition of (202) or (224) terms will not change the order parameter. C_{ij}, the strength of the (202) term, is proportional to the average polarisatility times the polarisability anisotropy. We consider equal-sized molecules and initially assume all cross-interaction potential coefficients to be given in terms of those of the pure components by the Lorentz-Berthelot rule (this is exact in the case of quadrupolar interactions, which allows for the vanishing of w_4 if the quadrupole moments of the two components have the same magnitude but opposite signs).

3. RESULTS

Fig. 2 shows a cross-section (taken at constant reduced quadrupole moment $Q/\sqrt{(C\sigma^5)}$ of the (x_2,Q^*) phase diagram of a binary LC mixture whose components differ by i) the sign (but not the strength) of their quadrupole moments), and ii) the fact that the surface favours homeotropic alignment of component 1 (V^1_{ext} < 0) and planar alignment of component 2 (V^2_{ext} > 0). x_2 is the concentration of component 2. B^*_v is a normalised combination of V_{ext} and $V(l_1 l_2 l)$.

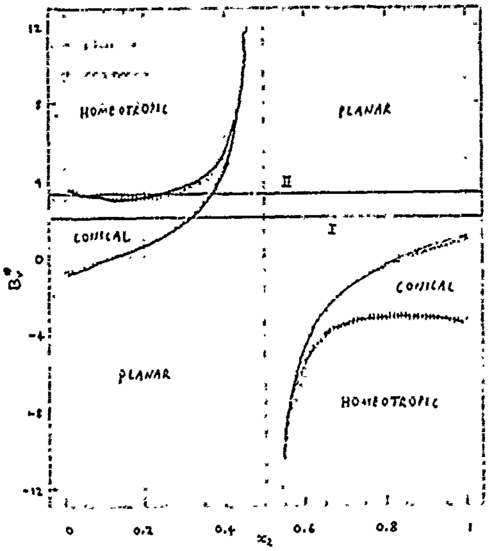


Figure 2: A cross-section of the (x_2,Q^*) phase diagram of the binary LC mixture described in the text for Q^* =1.0. Bv is the strength of the surface potential.

As $Q^* = Q_1^* = -Q_2^*$ increases, re-entrant conical anchoring becomes possible. In fig. 3 we plot the anchoring angle versus x_2 along trajectories I and II in fig. 2. The latter clearly displays anomalous, or re-entrant, behaviour.

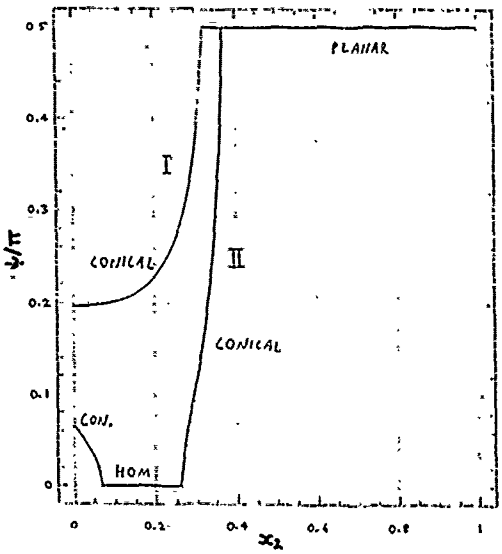


Figure 3: Tilt angle vs. x_2 along trajectories I and II in fig. 2.

4. CONCLUSIONS

We used a simple mean-field theory to derive a Landau-de Gennes expression for the surface free energy of a mixture of nematic LCs in contact with an anisotropic substrate. In spite of the fact that it contains a number of oversimplifications, the theory predicts fairly rich and interesting anchoring behaviour, including re-entrant conical anchoring, as a function of composition, when applied to a simple LC model.

REFERENCES

1. Mauguin, G. (1911). *Bull. Soc. Fr. Min.* 34, 71.
2. Grandjean, F. (1916). *Bull. Soc. Fr. Min.* 39, 164.
3. Cognard, J. (1982). *Mol. Cryst. Liq. Cryst. Suppl.* 1, 1.
4. Bechhoefer, J., Jerôme, B., and Pieranski, P. (1990). *Phys. Rev. A* 41, 3187.
5. Pieranski, P., and Jerôme, B. (1989). *Phys. Rev. A* 40, 317.
6. Pieranski, P., Jerôme, B., and Gabay, M. (1990). *Mol. Cryst. Liq. Cryst.* 179, 285.
7. Sullivan, D. E. (1985). Unpublished.
8. Sen, A. K. and Sullivan, D. E. (1987). *Phys. Rev.A* 35, 1391.
9. Tjipto-Margo, B. and Sullivan, D. E. (1988). *J. Chem. Phys.* 88, 6620.
10. Telo da Gama, M. M. (1984). *Mol. Phys.* 52, 611.
11. Sluckin, T. J., and Poniewierski, A. (1986) in *Fluid Interfacial Phenomena*, edited by C. A. Croxton, Wiley, New York.
12. Rapini, A. and Papoular, M. (1969). *J. Physique Colloq.* 30, C4.
13. Gray, C. G. and Gubbins, K. E. (1984). *Theory of Molecular Fluids*, vol. 1, Clarendon Press, Oxford.
14. Thurtell, J. H., Telo da Gama, M. M. and Gubbins, K. E.(1985). *Molec Phys.* 54, 321.

SYMBOLIC COMPUTATIONS FOR LIQUID CRYSTALS: INTEGRITY BASES

LECH LONGA[†,‡] AND HANS-RAINER TREBIN[‡]

[†]Jagellonian University,
Institute of Physics,
Reymonta 4, Krakow, Poland

[‡]Institut für Theoretische
und Angewandte Physik,
Pfaffenwaldring 57, Stuttgart, BRD

Abstract - Using algebraic processors general properties of SO(3) - invariant free energy expansion of biaxial liquid crystals are studied. This is achieved by decomposing invariants in terms of integrity bases as proposed by Judd et.al., Gaskell et.al., and Bistricky et.al. With the help of this approach a rigorous continuum free energy of general biaxial nematics is analyzed as an expansion in components of a symmetric and traceless tensor order parameter field $Q_{\alpha\beta}$ and its derivatives $\partial_\mu Q_{\alpha\beta}$. Next, a general theory of flexopolarization and a classification of local, polar structures in biaxial systems is offered. Finally, some consequences for biaxial smectic systems are summarized.

I. INTRODUCTION

Algebraic processors, like Macsyma or Mathematica, have been designed to perform symbolic, numerical and graphical calculations easily and to arbitrary precision. One can also use these processors to prepare input for, or analyse output from other external programs. Below we would like to summarize the results for continuum theory of biaxial liquid crystals obtained by combining symbolic and numerical possibilities of Macsyma.

A. Order Parameters

Orientational properties of liquid crystals are described in terms of irreducible spherical tensor fields $Q^{(L)}(r)$ of angular momentum L and of components $Q^{(L)}_m(r)$ [1]. Out of them the most important is L=2 quadrupole tensor which describes anisotropic part of electric- or magnetic susceptibilities. In cartesian representation $Q^{(2)}$ is identified with a second - order, symmetric and traceless tensor field Q(r) of components $Q_{\alpha\beta}(r)$. For polar liquid crystals additionally $Q^{(1)}_m(r)$ field must be retained.

B. Integrity bases

Theoretical studies of physical properties of the systems described in terms of $Q^{(L)}$ are based on the nonequilibrium free energy density expansion around an isotropic phase i.e. $Q^{(L)} = 0$. This expansion is an SO(3) symmetric polynomial in the components of irreducible spherical tensors $Q^{(L)}$ and their derivatives. Consequently, an important problem to solve is that of determining all SO(3) invariants which are homogeneous polynomials in components $Q^{(L)}_m$ and derivatives $\partial_\mu Q^{(L)}_m$. One can prove that for a finite set of Q fields it is possible to construct a finite basis of invariant polynomials (integrity basis) such that all other invariants can be written as polynomials of these basic invariants. This very elegant group theoretical method has been applied to problems with SO(3) symmetry by Judd et.al.[2], Gaskell et.al.[3] and Bistricky et.al.[4].

For large L, calculations of the integrity basis elements and studies of their properties are nontrivial algebraic problems, which appear to be perfectly suited for algebraic processors [5-7] . Here we summarize some of the results, obtained for the L = 2 tensor field and for the L = 1 vector field. Some other examples are found in refs.[2-7].

II. ELASTIC AND FLEXOPOLARIZATION MODES OF BIAXIAL LIQUID CRYSTALS

Elastic free energy of biaxial liquid crystals is defined as an expansion in $Q_{\alpha\beta}$ and $\partial_\mu Q_{\alpha\beta}$, where only rirst and second order terms in derivatives $\partial_\mu Q_{\alpha\beta}$ are retained. Thus, the expansion contains SO(3)- symmetric invariants built up from the tensors $Q_{\alpha\beta}Q_{\gamma\delta}\cdots Q_{\rho\sigma}(Q_{\mu\nu,\eta})$ and $Q_{\alpha\beta}Q_{\gamma\delta}\cdots Q_{\rho\sigma}(Q_{\mu\nu,\eta})(Q_{\xi\tau,\zeta})$ obtained by means of contractions with the Kronnecker deltas and the Levi-Civita tensors.

Similarly, the deformation induced polarization of biaxial systems depends on $Q_{\alpha\beta}$ and $Q_{\alpha\beta,\gamma}$ at each point. Since the effect is linear in deformations the corresponding flexopolarization part of the free energy must include the class of all linearly independent SO(3)- symmetric invariants $P_\xi Q_{\alpha\beta}Q_{\gamma\delta}\cdots Q_{\rho\sigma}\times(Q_{\mu\nu,\eta})$, where P_ξ is the polarization field ($P_\xi \leftrightarrow P^{(1)} \equiv Q^{(1)}$).

Now, decomposing the invariants of $Q_{\alpha\beta}$, $Q_{\alpha\beta,\gamma}$ and P_α in the corresponding integrity basis, one finds that the most general free energy expansion to all powers of $Q_{\alpha\beta}$ and up to second order in $Q_{\alpha\beta,\gamma}$ contains 39 basic elastic modes [6], where three of them are chiral. These generalize the concept of the splay-, bend-, and twist deformations of the director field, introduced by Oseen, Zocher and Frank. The associated, temperature dependent elastic constants are analytical functions of $\mathrm{Tr}Q^2$ and $\mathrm{Tr}Q^3$.

Similarly, the general flexopolarization free-energy density of chiral biaxial liquid crystals is composed of 12 basic deformation modes [7]. These are multiplied by arbitrary polynomials in $\mathrm{Tr}Q^2$ and $\mathrm{Tr}Q^3$ which define temperature- and position dependent flexocoefficients.

In both cases simpler forms of the free energy densities are obtained from the general expansion by imposing additional symmetry restrictions on the field $\underline{\underline{Q}}$. Details are given in refs.[6,7].

III. POLAR STATES OF BIAXIAL LIQUID CRYSTALS

With the help of integrity basis one finds very convenient approach to study selection rules for broken symmetry states in an arbitrary Landau free energy expansion [7]. As an example we investigate the correlation between polar nematic states and the properties of the integrity basis for invariants composed of the components of $P^{(1)}$ and $Q^{(2)}$.

The integrity basis for invariants of two order parameters $P^{(1)}$ and $Q^{(2)}$ is composed of six invariants $I_{\alpha\beta}$, whose degrees of $P^{(1)}$ and $Q^{(2)}$ are α and β, respectively. The invariants in Cartesian representation can unambiguously be identified as [7]:

$$I_{02} = \mathrm{Tr}\underline{\underline{Q}}^2,$$

$$I_{03} = \mathrm{Tr}\underline{\underline{Q}}^3,$$

$$I_{20} = P_\alpha P_\alpha,$$

$$I_{21} = P_\alpha Q_{\alpha\beta} P_\beta,$$

$$I_{22} = P_\alpha Q^2_{\alpha\beta} P_\beta \; - \; \tfrac{1}{3}\,\mathrm{Tr}(\underline{\underline{Q}}^2)\; P_\alpha P_\alpha,$$

and

$$I_{33} = P_\alpha P_\beta P_\gamma \; \varepsilon_{\alpha\mu\nu}\; Q_{\mu\beta} Q^2_{\nu\gamma},$$

where

$$108\,(I_{33})^2 =$$

$$-\,54\; I_{02}\; I_{20}\; (I_{22})^2 \;+\; 54\; I_{02}$$

$$(I_{21})^2\; I_{22} - 9\,(I_{02})^2\,(I_{21})^2\; I_{20}$$

$$+\,2\,(I_{02})^3\,(I_{20})^3 - 36\; I_{03}\,(I_{21})^3$$

$$+\,108\; I_{03}\;\; I_{21}\;\; I_{22}\; I_{20}$$

$$-\,12\,(I_{03})^2\,(I_{20})^3 \;-\; 108\,(I_{22})^3.$$

Consequently , the Landau free energy expansion for polar nematics is a stable polynomial in $I_{\alpha\beta}$ and, at most, a linear function in I_{33}. One finds that in addition to the uniaxial and biaxial ferroelectric nematic phases which are generated by the first five invariants, there exists a biaxial chiral ferroelectric nematic phase, generated by the I_{33} term.

From the form of I_{33} it is clear that the following must be fulfilled for the possible existence of the chiral biaxial phase: i) chiral molecules with a large dipole moment component, perpendicular to the long molecular axis ii) large molecular biaxiality, probably of the same order as the one observed in thermotropic biaxial nematics.

Similar statements hold for the smectic - C^* phase. One finds that if the S_C* phase is stabilized due to the piezoelectric coupling between P and a density wave then it must be described as a biaxial, uniform spiral with, at least two nonvanishing commensurate harmonics. Since the polarization in the S_C* phase is perpendicular to the local director, additionally the biaxial piezoelectric coupling invariant I_{33} must vanish.

For nonzero value of the I_{33} invariant another phase with the S_C* symmetry may be more stable. In this phase, the polarization is not perpendicular to the local director. Intrinsic biaxiality of chiral liquid crystalline molecules is the driving force, stabilizing this phase.

The above predictions are in accordance with recent expectations that the non centrosymmetric biaxial molecules with negative dielectric anisotropy may be good candidates to form phases with local ferroelectric, biaxial nematic order.

A thorough discussion of the phase diagrams for the cases discussed above will be presented elsewhere.

REFERENCES

1. G.R.Luckhurst and G.W.Gray, ed., The molecular physics of liquid crystals (Academic Press, London, 1979).
2. B.R.Judd, W.Miller Jr., J.Patera and P.Winternitz, J.Math.Phys., 15, 1787, (1974).
3. R.Gaskell, A.Peccia and R.T.Sharp, J.Math.Phys., 19 727, (1978).
4. J.Bystricky, J.Patera and R.T.Sharp J.Math.Phys., 23, 1560, (1982). 982).
5. H.-R.Trebin, L.Longa and B.Salzgeber, Phys.Stat.Solidi(b), 144, 73, (1987).
6. L.Longa, H.-R. Trebin, Phys.Rev.A, 39(4), 2160, (1989).
7. L.Longa, H.-R. Trebin, Phys.Rev.A, 42(6), 3453, (1990).

SHEAR-FLOW INSTABILITIES IN LIQUID CRYSTALS[1]

MITCHELL LUSKIN
School of Mathematics
University of Minnesota
Minneapolis, MN 55455 USA

AND

TSORNG-WHAY PAN
Department of Mathematics
University of Houston
Houston, TX 77204

Abstract. We use the Ericksen-Leslie equations to investigate the stability of simple shear flow between moving parallel plates for non-flow-orienting nematic liquid crystals. We show numerically that as the velocity of the plate is increased the first instability can be either in the shear plane (tumbling) or out of the shear plane, depending on the material constants. We also present numerical results for the continuation of the out-of-plane solution from its bifurcation point.

1. ERICKSEN-LESLIE EQUATIONS

We consider simple-shear flow between parallel plates at a distance $2h$ apart which are parallel to the x-y plane. We assume that the upper plate at $z = h$ is at rest while the lower one at $z = -h$ moves with velocity $\mathcal{V}$ in the y direction. The state of the nematic liquid crystal is described by its velocity $\mathrm{v} = \mathrm{v}(x, y, z, t)$ and its anisotropic axis $\mathrm{n} = \mathrm{n}(x, y, z, t)$ where $|\mathrm{n}| = 1$.

We first investigate simple shear flows of the form

$$\mathrm{v} = (0, v(z,t), 0) \qquad \mathrm{n} = (0, \cos\theta(z,t), \sin\theta(z,t)). \tag{1.1}$$

For flows of the form (1.1) the Ericksen-Leslie equations are

$$\rho\frac{\partial v}{\partial t} = \frac{\partial}{\partial z}\left(g(\theta)\frac{\partial v}{\partial z} + m(\theta)\frac{\partial\theta}{\partial t}\right) \qquad -h \le z \le h \tag{1.2}$$

$$2\gamma_1\frac{\partial\theta}{\partial t} = 2f(\theta)\frac{\partial^2\theta}{\partial z^2} + \frac{\partial f(\theta)}{\partial\theta}\left(\frac{\partial\theta}{\partial z}\right)^2 - 2m(\theta)\frac{\partial v}{\partial z} \tag{1.3}$$

where

$$g(\theta) = \alpha_1\sin^2\theta\cos^2\theta + \frac{\alpha_5-\alpha_2}{2}\sin^2\theta + \frac{\alpha_6+\alpha_3}{2}\cos^2\theta + \frac{\alpha_4}{2};$$
$$m(\theta) = (\gamma_1 + \gamma_2\cos 2\theta)/2; \qquad f(\theta) = \kappa_1\cos^2\theta + \kappa_3\sin^2\theta;$$

ρ is the density; $\alpha_1, \dots, \alpha_6$ are the Leslie viscosities; $\kappa_1, \kappa_2, \kappa_3$ are the Frank elastic constants; and $\gamma_1 = \alpha_3 - \alpha_2$, $\gamma_2 = \alpha_6 - \alpha_5$. Thermodynamic inequalities imply that $g(\theta) > 0$ and $f(\theta) > 0$ for all θ and that $\gamma_1 > 0$ [4]. We shall only consider flows in the non-flow-orienting regime $\gamma_1 > |\gamma_2|$, so $m(\theta) > 0$ for all θ.

We utilize the "strong anchoring" condition for n, i.e.,

$$\theta(-h,t) = \theta(h,t) = \theta_p. \tag{1.4}$$

where $\theta_p = 0$ or $\pi/2$ and the "no-slip" boundary condition for v

$$v(-h,t) = \mathcal{V}, \qquad v(h,t) = 0. \tag{1.5}$$

For steady flow,

$$g(\theta)\frac{\partial v}{\partial z} = c$$

where c is an integrating constant for (1.2), and (1.6) can be used to eliminate $\partial v/\partial z$ from (1.3) to obtain

$$2f(\theta)\frac{\partial^2\theta}{\partial z^2} + \frac{\partial f(\theta)}{\partial\theta}\left(\frac{\partial\theta}{\partial z}\right)^2 - 2c\frac{m(\theta)}{g(\vartheta)} = 0,$$
$$\theta(-h) = \theta(h) = \theta_p.$$

2. STABILITY EQUATIONS

We have solved the linearized stability equations for (1.2), (1.3) given by the following eigenvalue problem for the perturbations $e^{i\lambda t}V(z)$ of $v(z)$ and $e^{i\lambda t}\Theta(z)$ of $\theta(z)$

$$\lambda\rho V = \frac{\partial}{\partial z}\left(\frac{\partial g}{\partial\theta}\frac{\partial v}{\partial z}\Theta + g(\theta)\frac{\partial V}{\partial z} + \lambda m(\theta)\Theta\right) \tag{2 1}$$

$$2\lambda\gamma_1\Theta = 2\frac{\partial}{\partial z}\left(f(\theta)\frac{\partial\Theta}{\partial z}\right) + 2\frac{\partial f}{\partial\theta}\frac{\partial^2\theta}{\partial z^2}\Theta + \frac{\partial^2 f}{\partial\theta^2}\left(\frac{\partial\theta}{\partial z}\right)^2\Theta$$
$$-2\frac{\partial m}{\partial\theta}\frac{\partial v}{\partial z}\Theta - 2m(\theta)\frac{\partial V}{\partial z}, \tag{2.2}$$

$$V(-h) = V(h) = 0 \qquad \Theta(-h) = \Theta(h) = 0. \tag{2.3}$$

Previous authors [2,7,8] have dropped the inertial term $\lambda\rho V$ in the linear momentum equation (2.1) and have set the integrating constant in (2.1) equal to 0 to obtain the equation

$$\frac{\partial g}{\partial\theta}\frac{\partial v}{\partial z}\Theta + g(\theta)\frac{\partial V}{\partial z} + \lambda m(\theta)\Theta = 0. \tag{2.4}$$

They then use (2.4) to eliminate $\partial V/\partial z$ in (2 2) and to obtain the Sturm-Liouville eigenvalue problem

$$2\lambda\left[\gamma_1 - \frac{m(\theta)^2}{g(\theta)}\right]\Theta = 2\frac{\partial}{\partial z}\left(f(\theta)\frac{\partial\Theta}{\partial z}\right) + 2\frac{\partial f}{\partial\theta}\frac{\partial^2\theta}{\partial z^2}\Theta$$
$$+\frac{\partial^2 f}{\partial\theta^2}\left(\frac{\partial\theta}{\partial z}\right)^2\Theta - 2\frac{\partial m}{\partial\theta}\frac{\partial v}{\partial z}\Theta + 2\frac{m(\theta)}{g(\theta)}\frac{\partial g}{\partial\theta}\frac{\partial v}{\partial z}\Theta, \tag{2.5}$$

$$\Theta(-h) = \Theta(h) = 0. \tag{2.6}$$

If the "no-slip" boundary conditions are replaced by the boundary conditions

$$g(\theta(-h))\frac{\partial v}{\partial z}(-h) = c_1, \qquad v(h) = 0, \tag{2.7}$$

where c_1 is the given shear stress on the bottom plate, then the integrating constant in (2.4) is 0 and the linearized stability is given by (2.5), (2.6).

For $\lambda = 0$ the equations (2.1)-(2.3) are the equations defining a turning point for stationary solutions of (1.2)-(1.5) parametrized by the plate velocity $\mathcal{V}$ whereas for $\lambda = 0$ the equations (2.5), (2.6) are the equations defining a turning point for the stationary solutions of (1.2)-(1.4) parametrized by the boundary shear stress c_1 in (2.7).

We have computed liquid crystal flows of the more general form

$$\mathrm{v}(z,t) = (u(z,t), v(z,t), w(z,t)) \tag{2.7}$$
$$\mathrm{n}(z,t) = (\cos\phi(z,t), \sin\phi(z,t)\cos\theta(z,t), \sin\phi(z,t)\sin\theta(z,t)).$$

using the Ericksen-Leslie equations. Since the flow is incompressible, $w = 0$. The linearized stability equations for the Ericksen-Leslie equations for steady flows of the form (1.1) with respect to flows of the form (2.7) are given by the eigenvalue problem (2.1)-(2.3) for in-plane perturbations $e^{i\lambda t}V(z)$ of $v(z)$ and $e^{i\lambda t}\Theta(z)$ of $\theta(z)$, and by a similar eigenvalue problem for the out-of-plane perturbations $e^{i\lambda t}U(z)$ of $u(z)$ and $e^{i\lambda t}\Phi(z)$ of $\phi(z)$ [5].

[1] This work was supported in part by the National Science Foundation and the Air Force Office of Scientific Research through grants DMS 835-1080 and DMS 871-8881, the Army Research Office through grants DAAL03-88-K-0110 and DAAL03-89-G-0081, and by the Minnesota Supercomputer Institute.

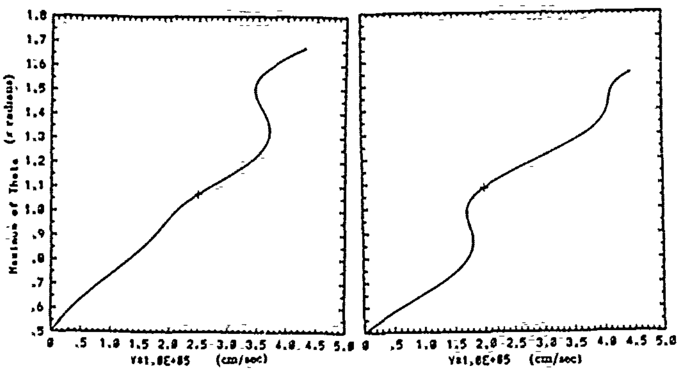


Fig. 1. At $\epsilon = .5$ the out-of-plane instability occurs first. At $\epsilon = .86$, the in-plane instability occurs first.

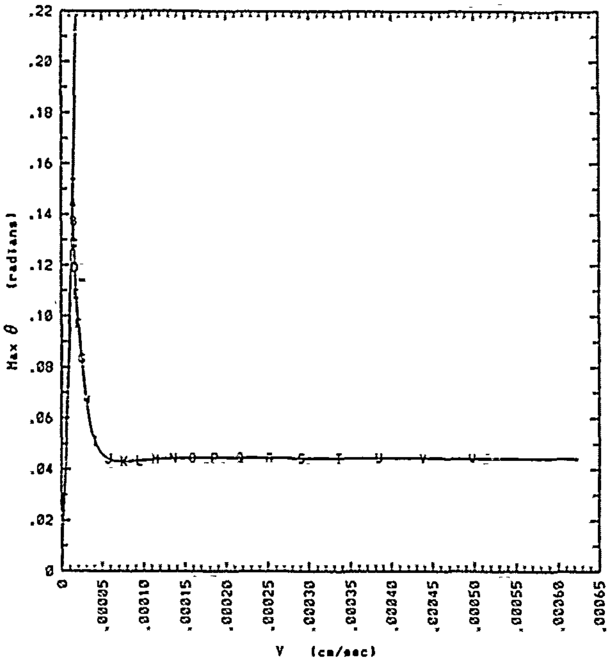


Fig. 2. Out-of-plane solution branch

3. COMPUTATIONAL RESULTS

Computational results for solution branches were obtained using the software package AUTO [3]. Our computational results show that the first instability can be in-plane or out-of-plane, depending on the material constants [5]. We model the behavior of 8CB just above the smectic A-nematic transition temperature with $\theta_p = \pi/2$ by the following material constants for $\epsilon > 0$:

$$\alpha_1 = 12\epsilon|\alpha_2|, \quad \alpha_2 = -.7, \quad \alpha_3 = \epsilon|\alpha_2|,$$
$$\alpha_4 = .58, \quad \alpha_5 = .7, \quad \alpha_6 = \epsilon|\alpha_2|,$$
$$\kappa_1 = 1.41 \times 10^6, \quad \kappa_2 = 1.023\epsilon\kappa_1, \quad \kappa_3 = 2.605\epsilon\kappa_1, \quad \rho = 1,$$

where viscosities are in poise, elastic constants are in dyne, density is in g/cm^3, and $h = 1$ cm. In Figure 1 above, the first in-plane instability occurs at the first turning point and the position of the first out-of-plane instability is marked by "+."

Next, we present computational results for the out-of-plane solution branch of the form (2.7) which has been continued from the bifurcation point for 8CB at 35° C [5]. These computations used the boundary data $\theta(-h) = \theta(h) = 0, \phi(-h) = \phi(h) = \pi/2$. The position of the bifurcation point in Figure 2 is marked by "+." Profiles of the solution $u(z)$, $\phi(z)$, and $\theta(z)$ are given in Figure 3 at the points on the solution branch marked by A–W.

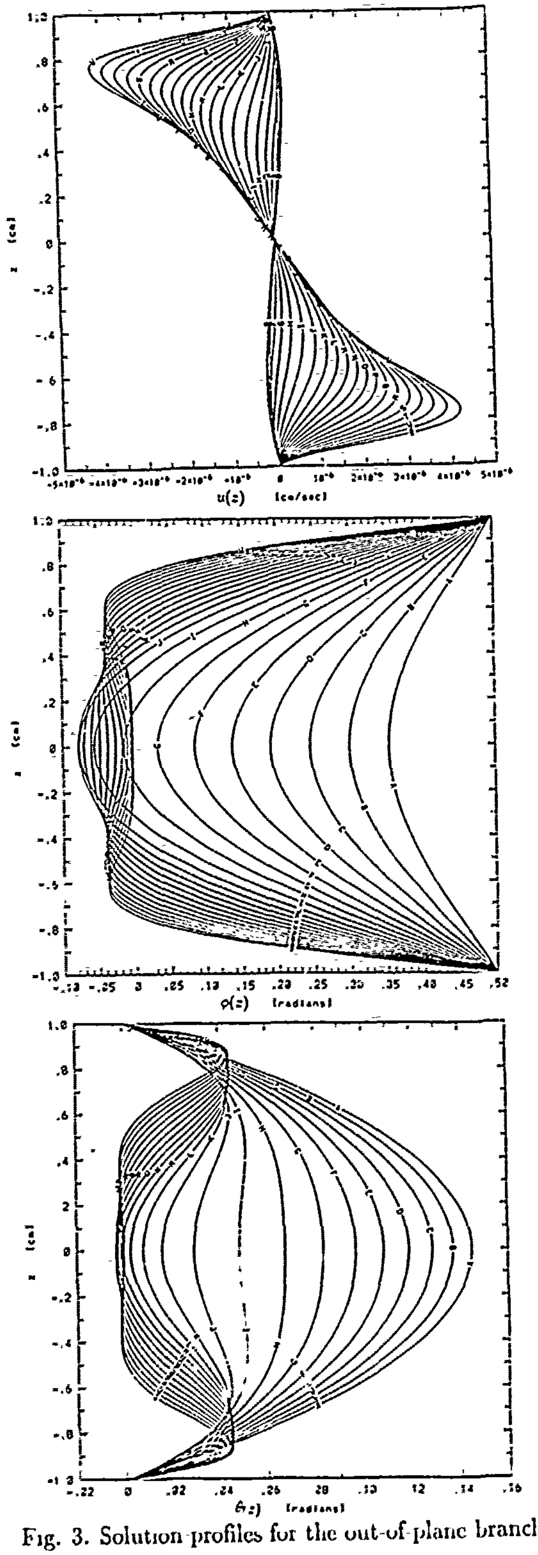


Fig. 3. Solution profiles for the out-of-plane branch.

References

1. P. Cladis and S. Torza, Phys. Rev. Lett 35 (1975), 1283.
2. P. Currie and G. MacSithigh, Q. J. Mech. Appl. Math. 32 (1979), 499.
3. E. Doedel, *AUTO*, 1986.
4. F. Leslie, Adv. Liq. Crystals 4 (1979), 1.
5. T.-W. Pan, Ph.D. Thesis, Univ. Minn. (1989)
6. P. Pieranski and E. Guyon, Comm on Physics 1 (1976), 45
7. I. Zúñiga and F. M. Leslie, Europhys. Lett. 9 (1989), 689.
8. I. Zúñiga and F. M. Leslie, Liq. Cryst. 5 (1989), 725.

HYDRODYNAMIC INSTABILITIES IN A NEMATIC LIQUID CRYSTAL DEVICE UNDER OSCILLATORY SHEAR

T. MULLIN
Clarendon Laboratory
Oxford

Abstract - We present the results of an experimental study of some hydrodynamic instabilities found in a large aspect ratio nematic liquid crystal device which is subjected to a linear oscillatory shear. The instabilities depend on shear rate, frequency and gap width of the cell. They also fall into two different basic classes for thick and thin cells.

1. INTRODUCTION

Hydrodynamic instabilities in thin homogenously aligned nematic liquid crystals which are subjected to linear oscillatory shear were first investigated by Clark, Saunders, Shanks and Leslie (1981). They found an instability in the form of I cm wide bands aligned in the direction of the shear which have alternating in-phase and out-phase twist separated by their dark regions. The layers were approximately 10μm thick and the frequency range was 10-100H_3. Thus the bands are large scale effects as the width of each is approximately 1000 layer thicknesses. In addition, Clark et al carried out a stability analysis of the equations of motion and obtained good agreement between theory and experiment. One crucial assumption in their analysis is that the critical shear rate for the appearance of the instability is independent of frequency. This appeared to be borne out by their observations for thin layers.

Clark et al also observed the formation of mechanical Williams domains for shear rates above the first instability described above. These are the mechanical equivalent of the Williams domains found in electrohydrodynamic convection (see for example Blinov (1983)). They have length scales of the order of twice the layer thickness and are thus microstructures when compared with the bands described above. The mechanical Williams domains have recently been investigated by Kozhevnikov (1986) for normally orientated nematics with applied oscillatory shear frequencies in the range 10^2 - 10^5 H_3. Finally, Guazzelli (1990) has also observed mechanical Williams domains when an elliptical oscillatory shear is applied to thin layers of nematics.

In the present study we have extended the work of Clark et al to investigate the effects of layer thickness on the observed instabilities. We have found that the critical shear rates for the first appearance of the large scale instabilities are strongly dependent on frequency although there is some evidence for independence at high frequencies. Patches of Williams domains are the first instabilities to appear with increase of shear rate for layers thicker than ~ 25 μm whereas the banded structures arise first for thin layers in agreement with the results of Clark et al. Finally, a nonlinear interaction between the mean flow field and the Williams domains can re-orientate the rolls so that there is both a change in direction and length scale of the microstructure. This in turn is accompanied with a novel long term 'memory effect' in the device.

2. EXPERIMENT

The apparatus consisted of two optically flat (λ/5) glass blocks of dimensions 10×8×1 cm mounted on an INVAR frame and spaced using three micrometers which are accurate to ± 0.5 μm. The lower block is mounded on linear bearings and is connected to a large electromagnetic vibrator. The amplitude of the applied vibration is measured both by a linear displacement device and using the Michelson interferometer technique suggested by Ben-Yosef, Ginio and Weitz (1974).

The nematic liquid crystal used was 108S TNC which has a positive coefficient, α_3. This material was aligned using rubbed PVA on the glass surfaces. The Williams domains can be observed directly from the scattered light and in detail through a microscope. On the other hand the banded instability were observed using cross polars as alternate bands correspond to 2Π and -2Π twists separated by thin dark lines of zero twist.

RESULTS

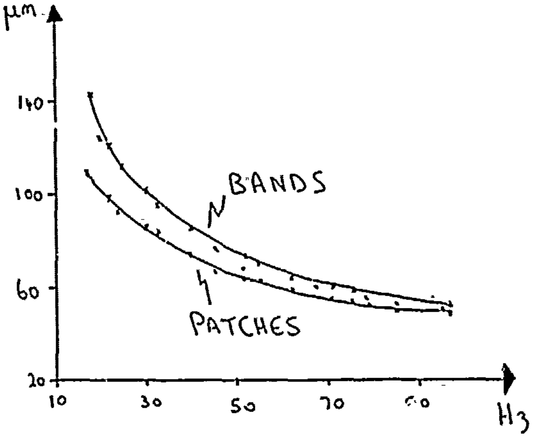


Fig. 1. Stability curves for a 30 μm cell. The upper curve shows the lower limit of stability of the band made as a function of frequency. The lower curve is for patches.

The results shown in figure 1 are the crical shear amplitudes plotted as a function of frequency for a 30 μm layer. The lower curve corresponds to the appearance of patches of Williams domains and the upper is for the banded structure. The estimate of the critical amplitudes are obtained by fixing the frequency and increasing the amplitude of the vibration in small steps until the instability is observed. Each instability has a rapid growth rate and so determination of a critical point is relatively straightforward. In addition, there is no evidence of any hysteresis in the transition and so the critical value can be determined by the appearance or disappearance of the instability.

The strong dependence of the critical amplitude on frequency for both instabilities is obvious, and their representative set is typical for all observations in the range 10-50 μm. For layers thinner than 20 μm the bands appear before the patches with increase in amplitude and vice versa for layers thicker than 25 μm. The exchange of priority between the two types of instability is an area of current investigation.

As observed by Clark et al the bands fill the whole cell and have the form of cm-wide stripes separated by thin dark lines and are orientated along the direction of the shear. The patches are generally directed in the same way but they n-ver fill the whole domain, i.e. the patches are separated by regions which have no apparent structure.

The number of bands or patches is depend-nt on the applied frequency and thicknes of the cell. Each of the states are repeatedly formed over a range of frequencies with quasistatic increase of amplitude. The exchange of priority between neighbouring states involves hysteresis and multiplicity of states, i.e. two different patch or band structures at the same supercritical parameter values. An example of this exchange process is sketched in figure 2 for the change over between two and three patches.

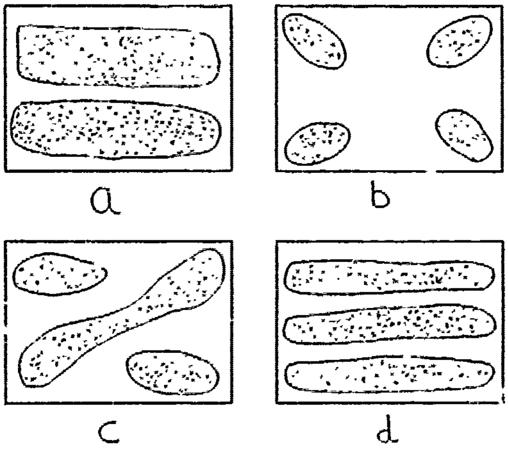


Fig. 2. Schematic of the exchange between two and three patches for a 50 µm cell (a) 15-45 H_3 (2 patches) (b) 45-75 H_3 (mixed mode) (c) 75-80 H_3 (mixed connected mode) (d) 80-100 H_3 (3 patches).

Over the lower frequency range, two patches are formed while at higher frequencies three are produced with slow increase in amplitude. The intervening range is covered by the interesting mixed mode behaviour. Starting from the low frequency end, four small patches are formed and one of the pair on a diagonal link across, swing through 45° and the other two small patches grow to form a three patch configuration. The process is symmetrical so that the growth along either diagonal is found.

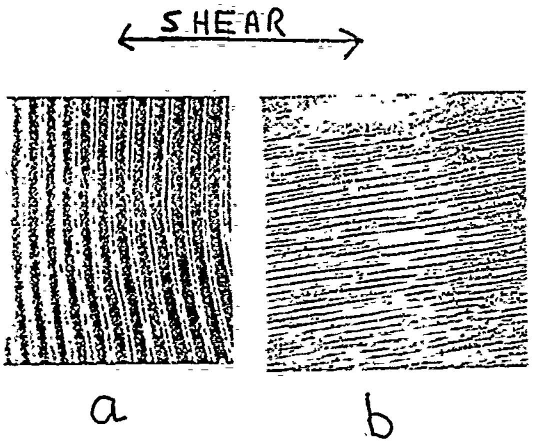


Fig. 3. Williams domains. (a) Normal to shear direction. (b) Aligned with shear: note edge of patch.

Finally, we show in figure 3 photocopies of photograhs of the two types of mechanical Williams domains formed in the patches. The ones shown in figure 3(a) are aligned at right angles to the shear and have a wavelength of approximately two layer thickness depending weakly on frequency. The second type shown in figure 3(b) are aligned in the direction of the shear and have a length scale approximately three times shorter than those shown in 3(a). They are formed i. the layer is left in a supercritical state for periods of minutes and seem to arise from an interaction with the weak mean flow field. Once areas of the cell have been reorientated like this then the orientation of the Williams domains becomes the preferred structure for periods of several hours.

Acknowledgements - The main part of this research is supported through the SERC 'Nonlinear Initiative'. GEC, Research at Hirst Research Centre provided helpful additional support. Finally, Mr. Nick Porter gave valuable assistance with some of the experimental work.

REFERENCES

1. Ben-Yosef, N., Ginio O. and Weitz, A. 'Measurement and analysis of mechanical vibrat[illegible] by means of optical heterodyning techniques', J. of Physics E, 7, (1974) pp.218-220.

2. [illegible]inov, L.M., 'Electro-optical and magneto-optical properties of [illegible] crystals', (John Wiley and Sons Ltd.) [illegible].

3. Clark, M.G., Saunders, F.C., Shanks, I.A. and Leslie, F.M. 'A study of flow alignment instability during rectilinear oscillatory shear of nematics', Mol.Cryst.Liq.Cryst. 70 (1981) pp.195-222.

4. Guazzelli, E. 'The motion of defects in convective structures of the elliptical shear instability of a nematic', (In 'Nematics' ed. (Ron, J.M., Ghidaglia, J.M. and Helein, F. NATO ASI Series C Vol. 332, 1990).

5. Kozhevnikov, E.N. 'Domain structure in a normally oriented nematic liquid crystal under the action of low frequency shear', Sov.Phys. JETP 64, (1986), pp.793-6.

DEFECT MEDIATED TRANSITIONS BETWEEN CONVECTIVE STATES IN A NEMATIC LIQUID CRYSTAL: EXPERIMENT AND SIMULATION [1]

JOETS ALAIN
Laboratoire de Physique des Solides
Université de Paris–Sud
91405 Orsay Cedex, France

Abstract- Hydrodynamical systems under constraint constitute rich examples of pattern-forming systems governed by nonlinear differential equations. The basic solutions may be either ordered states, or inhomogeneous states including (structural) defects. For instance, the convection in a nematic liquid crystal may exhibit a series of structures leading to chaos. We study here a direct transition between two structures of different symmetry and we show that the defects play an important role in the transformation. The complex evolution of the physical system is satisfactorily reproduced in numerical simulations using an appropriate nonlinear model of a so-called "amplitude equation".

I. INTRODUCTION

A nematic liquid crystal subjected to an AC electric field constitutes an example of a pattern-forming system that can exhibit ordered states as well as "complex states". As the amplitude of the applied field is continuously varied, the system undergoes a series of bifurcations (transitions) to ordered convective structures of decreasing symmetry: the Normal Rolls (NR), the Oblique Rolls (OR), Varicose, Bimodal, and finally the full chaotic state [1]. Each structure is characterized by a global symmetry. As is the case for every ordered structure, typical defects of the ordering exist and they appear as local states whereby the symmetry is broken. The topology of the defects is related to the broken symmetry. For instance, it is observed that the Normals Rolls have one type of defect: the (edge-) dislocation. It consists of an extra pair of rolls added to the structure at some point (the core).

II. TRANSITION FROM THE NORMAL ROLLS TO THE OBLIQUE ROLLS INDUCED BY THE DEFECTS

We have recently found a direct transition from the first structure, the Normal Rolls, to the second one, the Oblique Rolls, when two control parameters are simultaneously varied [2]. The evolution of the structure is first characterized by a growing undulation of the rolls along their axis (modulational instability). Then, dislocation pairs nucleate in the bending zones of the rolls, leading to the formation of small Oblique Rolls domains. As a consequence of the glide motion of the defects (motion perpendicular to the roll axis) the size of the OR domains grows and very large OR domains are finally obtained (OR structure). In this process, the role of the defects is clearly related to their complex dynamics, which acts as a local wavelength selection mechanism (local tilt of the rolls) and constitutes an efficient mechanism leading to a fast transition to the OR structure.

III. SIMULATION OF THE TRANSITION FROM THE NORMAL ROLLS TO THE OBLIQUE ROLLS

[illegible] this complex transition is actually untractable on the basis of the "microscopic equations". However, there exists an alternative approach based on the so-called amplitude equations [3,4,5]. This nonlinear model describes the evolution of the 2-D envelope of the rolls, supposed to vary slowly over large distances (with respect to the wavelength of the structure). A dislocation corresponds then to an isolated zero of the (complex) amplitude. For some values of the parameters in the model and of the initial state, we are able to simulate the observed NR → OR transition. In particular, we reproduce the initial modulational instability, as well as the nucleation of the defects, which by their glide motion are responsible of the formation of the OR domains.

IV. CONCLUSION

The convection in a nematic liquid crystal provides a very pertinent model for the study of extended nonlinear systems. The defects are not merely a local loss of order inside a well defined structure. They may play a very important role in the structural transition between two states of different symmetry. We show also that the "amplitude equations" are an efficient nonlinear simple model to simulate, at least qualitatively, such a complex behavior.

[1] This work was supported by the Direction des Recherches et Etudes Techniques under contract DRET/ERS/901616. The numerical simulations on a CRAY-2 computer were made possible by the Scientific Committee of the CCVR at Palaiseau (France).

[1] A. Joets and R. Ribotta. J. Physique (Paris), 47, 595 (1986).
[2] A. Joets and R. Ribotta. in "Nematics. Mathematical and Physical Aspects", J. M. Coron, J. M. Ghidaglia and F. Hélein editors, NATO ASI, Series C, 232, 189 (1991).
[3] A. C. Newell and J.A. Whitehead, J. Fluid Mech. 38, 279 (1969).
[4] W. Pesch and L. Kramer, Zeit. Phys. B 63, 121, (1986).
[5] E. Bodenschatz, W. Zimmermann and L. Kramer, J. Physique (Paris) 49, 1875 (1988).

ON DEFECTS IN NEMATIC LIQUID CRYSTALS

Epifanio G. VIRGA
Facoltà di Ingegneria
via Diotisalvi 2
56126 PISA, Italy

Abstract. *The classical theory of FRANK explains fairly well point defects that occur in the equilibrium configurations of nematic liquid crystals, but it fails to describe both disclinations and surface defects. ERICKSEN has recently proposed a model that aims to accomodate all defects in a unified theory. It is shown here by example how disclinations and surface defects fit into ERICKSEN's model. In both cases a defect arises when a material modulus attains a critical value.*

Liquid crystals are fluids that exhibit a preferred direction which varies in space : such a direction is the *optical axis*of the material. The orientation of the optical axis is customarily described by a unit vector field n. Here I shall conform to such a tradition, but I warn the reader that employing n to describe the orientation of the optical axis might be inappropriate, especially when one wishes to model general *defects*, that is, regions in space, of any dimension, where n suffers discontinuities. The orientation of the optical axis, that one is to represent by n, is unaffected by a reflection that changes n into -n everywhere. Thus, if n is reflected, say, across a surface, a fictitious defect arises since the optical axis is just the same on the two sides of the surface, while n suffers a jump.

A way to fix this up might be to describe the optical axis through the tensor field defined by

$$\text{(1)} \qquad \mathrm{N} := \mathrm{n} \otimes \mathrm{n}\ .$$

In general the fields n and N are not equivalent in the whole region where they are defined. We denote by E the three-dimensional Euclidean space and by V its translation space. Let B be the region of E occupied by the liquid crystal and let n be a field of B into S^2, the unit sphere of V. By (1) one readily associates to n a field N of B into the manifold N of all symmetric, rank-one tensor whose trace is 1. On the contrary, if a field N of B into N is given, in general one cannot retrace any field n of B into S^2 which is related to N through (1), without introducing fictitious defects.

For the problems I review here the fields n and N are globally equivalent. Thus, I will stay with n, though I hold that a description of the kinematics of liquid crystals in the manifold N wants.

The *degree of orientation* of a nematic liquid crystal is a scalar s that ranges in the interval [-1/2,1] . It is expained in Section 2 of [1] how the formal definition of s can be derived from the statistical distribution of molecular orientatations. Qualitatively, s measures at a macroscopic scale the degree of microscopic order, like n, it may vary in space The end-points of the interval [-1/2,1] represent two ideal situations. when s = -1/2, all the molecules, which resemble rods, are orthogonal to n, but do not lie in any preferred direction, when s = 1 all the molecules are parallel to each other. When s vanishes the orientation of the molecules is completely disordered. at a macroscopic scale, this means that the liquid crystal has become an isotropic fluid and n makes no sense at all.

When, besides n, also the degree of orientation s comes on the scene, the free energy per unit volume σ is allowed to depend on both s and its gradient, besides n and its gradient. ERICKSEN has discussed in [1] a general formula for σ; here we employ a special case of that formula:

$$\text{(2)} \qquad \sigma = \kappa\{k|\nabla s|^2 + s^2|\nabla \mathrm{n}|^2 + \psi(s)\}\ ;$$

both κ and $k$are positive constants, ψ is a function which describes a double-well potential having the local minimum at $s = 0$ and the absolute minimum at $s = s_0 > 0$. The main qualitative features of ψ are described, for examlpe, in Section 2 of [2].

Though there is enough evidence that ψ often prevails on the other terms of (2) (*cf.* [3], for example), I shall omit it in (2) and let its rôle to be played instead by a condition that sets $s = s_0$ wherever in ∂B n is prescribed.

When s is constant (2) becomes the *one-constant approximation* to FRANK's classical energy that has been widely studied FRANK's theory has been successful in solving a number of problems, but defects other than point defects escape its scope (see [4] and[5]).

On the other hand, line defects, also called *disclinations*, are often observed in ordinary liquid crystals, while surface defects occasionally occur in polymeric liquid crystals. The want for a unified treatment of defects prompted ERICKSEN to amend FRANK's theory. In the new theory the spatial changes in the degree of orientation prevent the free energy from being highly concentrated about defects; these are indeed to be identified with the regions where s vanishes: there the liquid crystal becomes isotropic and the optical axis has no meaning whatsoever. Thus, the new theory interprets defects as localized transitions of the liquid crystal to its isotropic phase, although no change in the temperature is involved.

The degree of orientation s and the optical axis n are delivered in B by the mappings

$$s : B \to [-1/2,1] \quad , \quad n : B\backslash S(s) \to S^2 \quad ,$$

where $S(s) := \{p \in B | s(p) = 0\}$ is called *singular set*. The points of the singular set are the only sites where defects of n actually occur (*cf.* [6]). Thus, if $S(s) \neq \emptyset$, we say that n is *singular*, otherwise we say that it is *regular*.

The total free energy is the functional

$$F[s,n] := \kappa \int_B \{k|\nabla s|^2 + s^2|\nabla n|^2\} \ .$$

The existence of minimizers for F and the degree of their regularity has been established by AMBROSIO (see[7] and [9]) and LIN (see [6] and [8]). Building upon their work, we consider now two variational problems for F whose solutions describe a disclination and a surface defect, respectively.

Let B be a circular cylinder and let D be the lateral boundary of B. We prescibe both n and s on D as follows

$$n = e_r \quad , \quad s = s_0 \quad \text{on } D \ . \tag{3}$$

These boundary conditions are axisymmetric, and so one would expect that the minimizers of F subject to (3) are axisymmetric too. Such a conclusion is in general false, as is shown in Section 5 of [10], but here one can prove that the minimizers of F within a quite broad class (which also includes non-axisymmetric fields) are indeed axisymmetric (see [11]). This variational problem was solved in [12]. There we reached the following conclusion:

If $k \leq 1$ then F subject to (3) attains its minimum when n is radial everywhere in B and s vanishes along the axis of B, where there is a disclination. If $k > 1$ then the field n that minimizes F subject to (3) is regular and closely resembles the celebrated solution of CLADIS & KLEMAN [13], which is fluted along the axis of the cylinder.

Let now B be the region of E between two square parallel plates. We call D^+ and D^-, respectively, the plates that bound B; we set $D = D^+ \cup D^-$. We prescribe n and s on D thus:

$$n = e_1 \text{ on } D^-, \ n = \cos\alpha_0 \, e_1 + \sin\alpha_0 \, e_2 \text{ on } D^+, \ s = s_0 \text{ on } D,$$

where $\alpha_0 \in \,]0,\pi[$ and e_1, e_2 are unit vectors directed along two adjacent sides of the plates. The variational problem for F subject to these boundary conditions was solved in [14]. Here is the conclusion we reached there.

If $k \leq (\alpha_0/\pi)^2$ then F attains its minimum when n is discontinuous at the mid plane between D^+ and D^-, and constant elsewhere. If $k > (\alpha_0/\pi)^2$ then the field n that minimizes F is regular and represents a non-uniform twist. Thus, for k sufficiently small a surface defect arises between two adjacent *domains*.

The effect of a potential like ψ on the occurrence of such a defect has been studied in [15] within a special class of minimizers: the qualitative features of the phenomenon seem to be unaffected by ψ, but the critical value of k decreases.

References

[1] J.L. ERICKSEN, *Liquid crystals with variable degree of orientation*, Arch. Rational Mech. Anal., 113(1991), 97-120.

[2] E.G. VIRGA, *Defects in nematic liquid crystals with variale degree of orientation*, in *Nematics* (J.M. CORON, J.M. GHIDAGLIA & F. HELEIN *Eds.*), Kluwer, Dordrecht, 1991, pp. 371-390.

[3] W.J.A. GOOSSENS, *Bulk, interfacial and anchoring energies of liquid crystals*, Mol. Cryst. Liq. Cryst., 124(1985), 303-311.

[4] R. HARDT, D. KINDERLEHRER & F.H. LIN, *Existence and partial regularity of static liquid crysta configurations*, Comm. Math. Physics, 105(1986), 547-570.

[5] R HARDT, D. KINDERLEHRER & F.H. LIN, *Stable defects of minimizers of constrained variational problems*, Ann. H. Poincaré, Anal. Nonlin., 5(1988), 297-322.

[6] F.H. LIN, *On nematic liquid crystals with variable degree of orientation*, to appear in Comm. Pure Appl. Math. (1991).

[7] L. AMBROSIO, *Existence of minimal energy configurations of nematic liquid crystals with variable degree of orientation*, Manuscripta Math., 68(1990), 215-228.

[8] F.H. LIN, *Nonlinear theory of defects in nematic liquid crystals; phase transion and flow phenomena*, Comm. Pure Appl. Math., 42(1989), 789-814.

[9] L AMBROSIO, *Regularity of solutions of a degenerate elliptic variational problem*, Manuscripta Math., 68(1990), 309-326.

[10] R. HARDT, D. KINDERLEHRER & F.H. LIN, *The variety of configurations of static liquid crystals, in Variational methods*, (BERESTYCKI, CORON & EKELAND *Eds.*), Birkhauser, Basel, 1990, pp. 115-131.

[11] V M TORTORELLI & E.G. VIRGA, *Axisymmetric boundary value problems for nematic liquid crystals with variable degree of orientation*, IMA Preprint Series, 1991.

[12] V.J. MIZEL, D. ROCCATO & E.G. VIRGA, *A variational problem for nematic liquid crystals with variable degree of orientation*, Research Report CMU, 1990.

[13] P.E. CLADIS & M. KLEMAN, *Non-singular disclinations of strength S=+1 in nematics*, J. Phys. (Paris), 33(1972), 591-598.

[14] L. AMBROSIO & E.G. VIRGA, *A boundary-value prolem for nematic liquid crystals with a variale degree of orientation*, to appear in Arch. Rational Mech. Anal., 1991.

[15] D. ROCCATO & E.G. VIRGA, *On surface defects in nematic liquid crystals with variable degree of orientation*, preprint, 1991.

4X4 MATRIX OPTICS THE "SLOW" WAY

DWIGHT W. BERREMAN
Fraunhofer I.A.F., 7800 Freiburg i.Br., Germany

Abstract - The 4X4 matrix method of computing reflection and transmission by layered structures can be very fast if a few simple mathematical and programming procedures are followed to enhance efficiency. Failure to utilize these procedures has led to a common misapprehension that the 4X4 matrix method is extremely time consuming. Structures with n-1 regular periodic variations parallel to the surface can be treated approximately using a 4nX4n extension of the method, to which most of these procedures for enhancing efficiency also apply.

INTRODUCTION

The 4X4 matrix method[1-3] is a generalization, for anisotropic media, of the Abelès[4] double-2X2 matrix method for computing reflectance and transmittance of plane, isotropic layered structures. Since the 4X4 method was first used to compute reflection and transmission in twisted liquid crystal structures[3] there have been continued attempts to find faster ways to solve the problem. However, the fact that many groups use the 4X4 method regularly to design twist- and super-twist cells with phase-retarding plates, color filters and polarizers all in the structure, attests to its practicality if it is efficiently programmed.

Closed-form solutions for isotropic materials[4] and for uniaxial and biaxial materials in certain special orientations have been known for many years.[1,5] It was also mentioned in several early publications[1,2,5] that the transfer matrix for a uniform slab of any thickness could be obtained through solving for the eigenvalues of the 4X4 differential transfer matrix, Δ, to obtain the propagation vectors of the four optic modes in the medium. The secular equation for these modes was given in compact form by Teitler and Henvis,[2] Eq.16. Wöhler et al,[6] recently published the solution to the secular equation in closed form, and the resulting elements for the 4X4 transfer matrix, for a uniform layer of uniaxial material of any thickness and any orientation. Use of these solutions has recently been termed the "fast" method.

Most nematic twist cells can be reasonably well approximated by using rather few, thick sub-layers of uniaxial liquid crystal. Most polarizers are also uniaxial, as are most retarding films. The "fast" method is advantageous in these regions.

If the liquid crystal in the cell is subdivided into very many short segments for high precision the "slow" method is faster than the "fast" because eigenvectors are not computed. If ferroelectric smectic displays and nematic displays with high field gradients are of interest, then biaxiality may be significant. The "fast" method is then tedious and slower.

Two misconceptions discourage people from trying the "slow" method, despite its simplicity and possible speed. The first is the supposition that it is necessary to divide the structure into segments of thickness T_m that are so thin that the square of the phase change of the wave across each segment is negligible; less than the wavelength in the medium, λ_m, divided by $2\Pi \cdot 10^4$. The second is that one must therefore multiply 4X4 matrices 62,800 times just to get through a thickness of one wavelength within the medium.

I had the first misconception at the outset of my work with T.J.Scheffer. However, I used the "method of repeated squaring" to greatly reduce the number of matrix multiplications. This method is still the answer to the second objection. Without any other improvements the number of matrix squarings required to get through λ_m would be about $\log_2(62,800) = 16$, rather than 62,800.

By using an exponential series expansion[1,4,5] adroitly, the thickness of "one step" may be increased to $\lambda_m/2\Pi$. For an accuracy of about one part in 10^8, six matrix multiplications are required in that expansion. The number of matrix multiplications required to get through λ_m is thus reduced to $6 + \log_2(2\Pi) = 9$, not 16, much less 62,800.

METHODS FOR MAKING EFFICIENT "SLOW" PROGRAMS

A: Fast Matrix Multiplication

Multiplication of two 4X4 matrices of complex numbers requires that the computer find the locations in storage of $16 \cdot 2 \cdot 2 = 64$ real numbers, then do $4 \cdot 4 \cdot 16 = 256$ multiplications of real number pairs, then $32 \cdot 3 = 96$ additions, and finally store the 32 sums. For the case of 8X8 matrix optics[7], the number of multiplications, which take most of the time, is 2048. Actual measurement confirms that matrix multiplication is the most time-consuming part of the fastest 4nX4n programs that we have made using the "slow" method, but when n=1, (4X4 matrix optics), it is usually only somewhat longer than the rest of the computation.

Many 4X4 multiplications are to be done even if individual transfer matrices are to be found by the "fast" method. Hence it is important, in any case, to write an efficient matrix multiplication program. With most compilers a great saving in time is achieved by writing out the 4X4 matrix product elements explicitly; avoiding loops and computed indexes.

B: Repeated Squaring Technique

Let T_m designate the thickest layer for which the transfer matrix can be computed in "one step" without making significant approximation errors. This thickness depends on the method used, as mentioned before. The next problem is to obtain the transfer matrix P(T) over a slab of thickness $T > T_m$ using the relation $P(T) = P(H)^j = P(jH)$ adroitly, where $H < T_m$. The fastest method is to set $P(T) = P(2^nH)$ = [P(H) squared n times], so that $j=2^n$. Thus $H = T/2^n$, where n is the smallest integer larger than $\log_2(T/T_m)$. Of course, if $T < T_m$ then $n = 0$, $H = T$, and no squaring is necessary.

C: Matrix Differential Equation and Exponential Series Expansion for a Thin Layer

The 4x4 matrix method starts with a matrix differential equation, equivalent to Maxwell's equations,[1-7] $d\Psi_i/dz = (2\Pi/\lambda)\Delta_{ij}\Psi_j$, where λ is the vacuum wavelength. Contraction over repeated indexes is implied. Δ is a 4X4 differential matrix whose elements depend only on the direction of propagation of the incident beam and the possibly complex optical dielectric tensor, ϵ, at that level in the multilayer. (The principal values of ϵ are the squares of the principal refractive indexes, n_p+ik_p). The four-element "vector" Ψ contains the four electric and magnetic field components parallel to the surface of the layered structure, or normal to the "z" axis.

Let S be the sine of the angle of incidence of light from vacuum, and assume that ϵ is symmetric.

Then the differential matrix is $\Delta = iD/e_{zz} =$

$$\frac{i}{e_{zz}} \begin{vmatrix} -Se_{xz} & e_{zz}-S^2 & -Se_{yz} & 0 \\ e_{xx}e_{zz}-e_{xz}^2 & D_{11} & e_{xy}e_{zz}-e_{xz}e_{yz} & 0 \\ 0 & 0 & 0 & 1 \\ D_{23} & D_{13} & (e_{yy}-S^2)e_{zz}-e_{xz}e_{yz} & 0 \end{vmatrix}$$

and $\Psi = \begin{vmatrix} E_x \\ H_y \\ E_y \\ -H_x \end{vmatrix}.$

We compute the P-matrix from the last surface to the first. This avoids the necessity of doing a matrix inversion at the end, which saves significant computer time and programming effort. If z increases in going from the entrance to the exit side of the layered structure then the solution to the matrix differential equation in a region of invariant optical dielectric tensor of thickness $H < T_m$ is[1,4,5]

$$\Psi(Z-H) = P(-H):\Psi(Z) = [1 - h\Delta + (h\Delta)^2/2! - (h\Delta)^3/3! + (h\Delta)^4/4! - \cdots]:\Psi(Z),$$

where $h = 2\pi H/\lambda$. $P(-H)$ is the transfer matrix relating the field Ψ at the side of the layer where the beam enters to that at the exit side, and $h^n\Delta_{ik}\Delta_{kl}\cdots\Delta_{nj}$ defines the ij element of $(h\Delta)^n$. Accuracy of one part in 10^8 may be achieved if no element of $h\Delta$ has magnitude larger than about unity, and if the series runs to $(h\Delta)^{11}/11!$.

To find an appropriate value of h for a uniform layer of thickness T, first find the square of the matrix, $\Delta_{ik}\Delta_{kj}$. (This squared matrix is used again in the two series to follow, so it is not wasted.) Find the absolute square of $\Delta_{1k}\Delta_{k1}$ and of $\Delta_{3k}\Delta_{k3}$ and select the larger. This number is the magnitude of the largest element likely to appear in Δ^4. Call its fourth root Δ_m. The series will converge rapidly if $h\Delta_m < 1$. Count the number of times, n, that the layer thickness T must be halved before $H = T/2^n < \lambda/(2\pi\Delta_m)$ Then n is the number of times P(H) must be squared to get P(T), and $h = 2\pi H/\lambda$.

To achieve the eleventh order expansion with six matrix multiplications, write

$$P(-H) = 1 + (h\Delta)^2/2! + (h\Delta)^4/4! + (h\Delta)^6/6! + (h\Delta)^8/8! + (h\Delta)^{10}/10!$$

$$-h\Delta\cdot[1 + (h\Delta)^2/3! + (h\Delta)^4/5! + (h\Delta)^6/7! + (h\Delta)^8/9! + (h\Delta)^{10}/11!].$$

If $T < \lambda/(2\pi\Delta_m) = T_m$, then the preceding two series may be truncated. Accuracy of one part in 10^8 is maintained when the tenth power terms are neglected if $\log_2(T/T_m) < -0.9$; the eighth power if < -2.0; and the sixth if < -7.5.

If the medium does not absorb light, then e is a real tensor or scalar and Δ is then purely imaginary. Hence Δ^2 is a matrix of real numbers. Additional speed may then be obtained in such media by writing a separate program to multiply real 4X4 matrices. Such a program is nearly four times faster than one for complex numbers. However, the special program is useful only for expanding the power series, not for subsequent repeated squaring.

D: Abelès' Method for Isotropic Regions

For isotropic media, $e_{xx}=e_{yy}=e_{zz}=e=(n+ik)^2$ and $e_{xy}=e_{yz}=e_{zx}=0$. Hence the local 4X4 matrices, Δ and P(T), have zeros in the off-diagonal 2X2 quadrants. The 4X4 matrix is then composed of Abelès'[5] two 2X2 matrices along the diagonal:

$$\Delta = i\cdot\begin{vmatrix} 0 & 1-(S^2/e) & 0 & 0 \\ e & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & e-S^2 & 0 \end{vmatrix}.$$

The square, $h^2\Delta_{ij}\Delta_{kj}$ is just a constant, (complex if the medium absorbs light), multiplied by the unit 4X4 matrix. That constant is

$$h^2(\Delta_{12}\ \Delta_{21}) = h^2(\Delta_{34}\ \Delta_{43}) = h^2(e - S^2).$$

The two series reduce to power series in this (possibly complex) number. The first series generates the cosine and the second (except for a factor) the sine of the number. These are the Abelès closed-form solutions. No matrix multiplication is required to obtain the 4X4 transfer matrix in this case, but the magnitude of the constant should be less than unity, as in the more general case of the 4X4 matrix.

One may prefer to use pre-programmed functions for the complex sines and cosines, which may or may not take account of the slow convergence problem for large arguments.

There is nothing comparable to Abelès' method for 8X8 or higher-order matrix optics. The possibility of a skew angle between the incident beam and the direction of the lateral periodic structure leaves open the possibility of non-zero elements in any region of the matrix.

CONCLUDING REMARKS

We find that each computation of reflectance and transmittance by a supertwisted nematic liquid crystal cell takes 1.53 seconds using only the "slow" method on a 386-series 16-bit 16-MHz PC with math co-processor, when the following parameters are used: The cell has two 1-mm glass cover plates, two absorbing uniaxial polarizers, two absorbing InSb oxide conductive layers, a uniaxial compensating fractional-wave plate, and the non-absorbing liquid crystal is subdivided into 36 layers. The "fast" method would enhance the speed somewhat in going through the polarizers and retarding plate, but less in going through the much-subdivided liquid crystal.

If one had a computer with an array processor, computing the transfer matrix for a uniform slab might be about as fast as computing a sine, cosine or exponential on a single-processor computer. In that case the "slow" method might be somewhat faster than the "fast" method of 4X4 matrix optics, especially for biaxial materials, since the eigenvector problem would be avoided at each different layer. However, an array processor could probably be used to better advantage in liquid crystal cell simulation to run several different wavelengths or directions of incidence simultaneously. For 8X8 and higher order matrix diffraction problems the potential for faster matrix multiplication with an array processor might enhance the speed considerably.

REFERENCES

1) D.O.Smith, Optica Acta 12,13(1965)

2) S.Teitler & B.Henvis, J.Opt.Soc.Am.60,830(1970)

3) D.W.Berreman & T.J.Scheffer, Phys.Rev.Lett.25,577(1970)

4) Florin Abelès, Ann. de Physique 5,596(1950)

5) D.W.Berreman, J.Opt.Soc.Am.63,502(1972)

6) H.Wohler, G.Haas, M.Freisch & D.A.Mlynski, J.Opt.Soc.Am.A5,1554(1988)

7) D.W.Berreman & A.T.Macrander, Phys.Rev.B37,6030(1988)

Director Configurations and Optical Properties of Twisted Nematic Layers with Weak Anchoring in the Tilt and Twist Angle

Hirning R., Funk W., Trebin H.-R.
Institut für Theoretische und Angewandte Physik der Universität Stuttgart,
Pfaffenwaldring 57, W-7000 Stuttgart 80, F.R.G.

Abstract

Numerical calculations of director configurations and electrooptical characteristics in symmetrical and nonsymmetrical nematic layers are presented, when weak anchoring in the tilt *and* twist angle of the director is assumed. In cells with bistabilities we investigate the influence of the anchoring parameters and device parameters on the width of the hysteresis. Using the 4 x 4-matrix-formalism of Berreman, we demonstrate the influence of the weak anchoring on the transmission vs. voltage characteristic and CIE color coordinates.

Our main emphasis was to implement an efficient code which works properly and fast over a broad range of material and device parameters.

1 Introduction

The director configuration in twisted nematic layers like TN[1]-, OMI[2]- or SBE[3]-cells is determined by the following three major features: first, the elastic forces in the liquid crystal, described by the well known Frank-Oseen-Zocher free energy density. Second, the influence of an external applied voltage modeled through an electric energy term of the form $\frac{1}{2}\vec{D}\vec{E}$, where $\vec{D}$ is the displacement vector and $\vec{E}$ the internal electric field vector, and third, the anchoring of the director at the substrate boundaries of the layer. Recently, some experimental studies of the anchoring have been published[4, 5, 6], showing that typical values of the anchoring energy are in the range 10^{-6}N/m to 10^{-5}N/m for homeotropically anchored nematics and 10^{-6}N/m to 10^{-3}N/m for planarly oriented nematics. In the last case, the twist anchoring energy is one order of magnitude smaller than the tilt energy.

In the following, the two kinds of anchoring are combined and studies are presented of the influence on the hysteresis width and on the electrooptical properties in symmetrical as well as in nonsymmetrical cells.

2 Theory

We consider a nematic cell of thickness d located between the planes $z = 0$ and $z = d$ of a Cartesian coordinate system. The director $\vec{n}$ is described by the tilt-angle θ (measured from the layer normal) and the twist angle φ. The dielectric constants are denoted by $\epsilon_\parallel$ and $\epsilon_\perp$. The elastic constants for splay, twist and bend are denoted by k_{11}, k_{22} and k_{33} respectively. The pitch of the material induced through a chiral dopant is named p_0.

Using the abbreviations

$$\begin{aligned} a_1 &= k_{33}\cos^2\theta + k_{11}\sin^2\theta \\ a_2 &= (k_{33}\cos^2\theta + k_{22}\sin^2\theta)\sin^2\theta \\ a_3 &= \frac{2\pi}{p_0}k_{22}\sin^2\theta \\ a_4 &= \varepsilon_0(\epsilon_\perp + \Delta\epsilon\cos^2\theta), \end{aligned} \tag{1}$$

the free energy density in the bulk can be written as:

$$f_B = \frac{1}{2}a_1\theta'^2 + \frac{1}{2}a_2\varphi'^2 - a_3\varphi' - \frac{1}{2}a_4\Phi'^2. \tag{2}$$

where the prime indicates differentiation with respect to z and where Φ is the electric potential inside the layer. Note that the last term represents the electric contribution $\frac{1}{2}\vec{D}\vec{E}$, when we assume a dielectric material law for uniaxial nematics in the form:

$$D_i = (\epsilon_\perp\delta_{ij} + \Delta\epsilon\, n_i n_j)E_j, \tag{3}$$

with $E_j = -\partial_j\Phi$.

The weak anchoring in the tilt and twist angle at the top and bottom of the cell is described by a surface free energy of the Rapini-Papoular type[7]:

$$\begin{aligned} F_S^0 &= \tfrac{1}{2}C_\theta^0\sin^2(\theta(0) - \theta_p^0) + \tfrac{1}{2}C_\varphi^0\sin^2(\varphi(0)) \\ F_S^d &= \tfrac{1}{2}C_\theta^d\sin^2(\theta(d) - \theta_p^d) + \tfrac{1}{2}C_\varphi^d\sin^2(\varphi(d) - \varphi_p) \end{aligned} \tag{4}$$

The factors $C_\theta^{0,d}$ resp. $C_\varphi^{0,d}$ measure the anchoring strength in the tilt and twist angle respectively, $\theta_p^{0,d}$ (pretilt) describe the preferred tilt angle of the director at the surfaces, φ_p (pretwist) is the difference between the preferred orientations in the twist angle at the top and bottom surface. The influence of the surface is restricted to the place of the aligning substrate.

The total free energy per unit area of the cell is now given by:

$$F/A = \int_0^d f_B\,dz + F_S^0 + F_S^d \tag{5}$$

3 Numerical Procedure

The first step in the calculation of optical properties consists in the determination of the director configuration. To this end, we transform the Euler-Lagrange equations resulting from the extremalisation of the bulk free energy into a system of Hamilton equations by performing a Legendre-transformation with respect to the variables θ, φ and Φ:

$$\begin{aligned} \theta' &= \frac{\partial f_B^L}{\partial T} = \frac{T}{a_1} \\ \varphi' &= \frac{\partial f_B^L}{\partial P} = \frac{P + a_3}{a_2} \\ \Phi' &= \frac{\partial f_B^L}{\partial U} = -\frac{U}{a_4} \\ T' &= -\frac{\partial f_B^L}{\partial\theta} = \frac{b_1}{2a_1^2}T^2 - \frac{b_3}{a_2}(P + a_3) + \frac{b_2}{2a_2^2}(P + a_3)^2 - \frac{b_4}{2a_4^2}U^2 \\ P' &= -\frac{\partial f_B^L}{\partial\varphi} = 0 \\ U' &= -\frac{\partial f_B^L}{\partial\Phi} = 0, \end{aligned} \tag{6}$$

where f_B^L is the Legendre transform of f_B:

$$f_B^L(\theta, T, P, U) = \theta'T + \varphi'P + \Phi'U - f_B. \tag{7}$$

and $b_i = \frac{\partial a_i}{\partial\theta}$, $i = 1,4$, and T, P, U are the conjugated momenta:

$$\begin{aligned} T &= \frac{\partial f_B}{\partial\theta'} = a_1\theta' \\ P &= \frac{\partial f_B}{\partial\varphi'} = a_2\varphi' - a_3 \\ U &= \frac{\partial f_B}{\partial\Phi'} = -a_4\Phi'. \end{aligned} \tag{8}$$

As φ and Φ are cyclic variables, the corresponding momenta P and U represent integration constants for the problem. To these equations we have to add the boundary conditions expressed in the new variables at $z = 0$:

$$\begin{aligned} T &= +\frac{\partial F_S^0}{\partial\theta} = C_\theta^0\sin(\theta - \theta_p^0)\cos(\theta - \theta_p^0) \\ P &= +\frac{\partial F_S^0}{\partial\varphi} = C_\varphi^0\sin\varphi\cos\varphi, \end{aligned} \tag{9}$$

and the corresponding equations at $z = d$.

The equations (6) and (9) represent a nonlinear boundary value problem, which we solve numerically by a multidimensional shooting method using standard library routines.

The second step is to solve the Maxwell equations inside the layer. For this problem we use the 4×4 formalism of BERREMAN[8, 9].

On a IBM PC/AT with 10 MHz, for instance, the calculation of an electrooptical characteristic needs about ten minutes independent of the existence of bistabilities.

4 Results

4.1 Director Configurations

In fig. 1, we show director configurations for a symmetrical and nonsymmetrical 90°-TN-cell with $\theta_p^d = 89°$ and $C_\theta^{0,d} = 10^{-4}$N/m in both cells; C_φ^d is 10^{-4}N/m whereas C_φ^0 is 10^{-5}N/m for the nonsymmetrical cell. The parameter for the curves is the applied voltage.

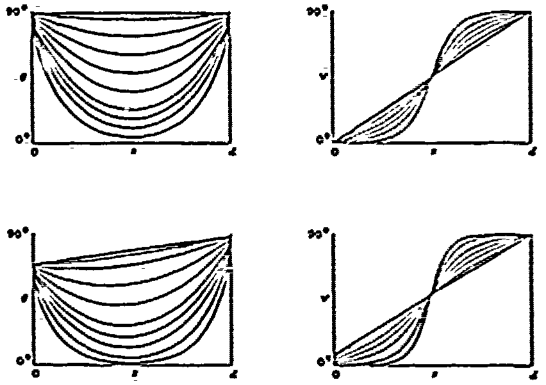


Figure 1: Director configurations for a symmetrical (top) and for a nonsymmetrical (bottom) TN-cell with $\theta_p^0 = 69°$; material parameters are $\frac{k_{33}}{k_{11}} = 0.885$, $\frac{k_{22}}{k_{11}} = 0.438$, $\varepsilon_\parallel = 8$, $\varepsilon_\perp = 3.5$; the cell thickness is 5.6μm.

With increasing voltage, the director tries to align parallel to the applied electric field, e.g. the tilt angle decreases, whereas the twist becomes more and more nonlinear.

4.2 Width of the Hysteresis

In highly-twisted cells with nonzero pretilt, there is the possibility of bistable director configurations[10], which becomes evident by a hysteresis in the θ_m vs. voltage curve. In fig. 2, we have plotted the width of the hysteresis ΔV for a symmetrical cell as a function of C_φ and φ_p for fixed $C_\theta = 10^{-4}$N/m and $\theta_p = 15°$; p_0 is chosen in such a way, that it matches φ_p. We get a monotonic increase of ΔV in φ_p. Further, it can be seen that for fixed φ_p, the width of the hysteresis falls with decreasing C_φ.

4.3 Transmission vs. Voltage Curves, Color Coordinates

One of the most important features for a twisted nematic layer is its transmission vs. voltage characteristic. In fig. 3 (left), we show these curves for a 90°-TN-cell for which the surface parameters are the same as in section 4.1, except that we vary θ_p^0.

It is seen, that an increase in the asymmetry of the cell decreases the optical threshold voltage.

The color coordinates of the cell with the applied voltage as parameter are shown in the right of fig. 3. The cell with the greatest asymmetry gives smallest color changes.

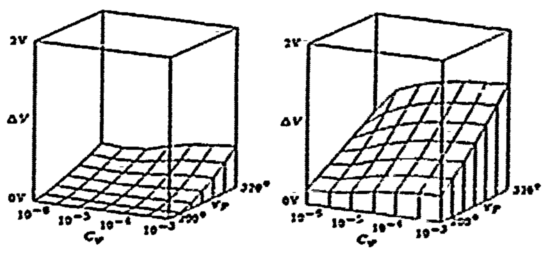


Figure 2: Hysteresis width; Material parameters are $\frac{k_{33}}{k_{11}} = 2.1$, $\frac{k_{22}}{k_{11}} = 0.4$, $\varepsilon_\parallel = 20$, $\varepsilon_\perp = 10$; the cell thickness is 5.6μm.

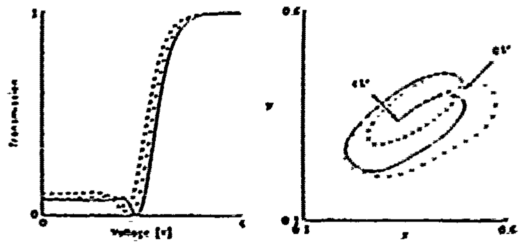


Figure 3: Left: Transmission vs. voltage characteristic the optical birefringence indices are $n_o = 1.5$ and $n_e = 1.65$, polarizer and analyzer are parallel; the solid line represents the curve for $\theta_p^0 = 89°$, the dashed-dotted line for $\theta_p^0 = 79°$, the dashed line for $\theta_p^0 = 69°$.
Right: Corresponding color coordinates

References

[1] Schadt M., Helfrich W., Appl. Phys. Lett. 18, 127 (1971)

[2] Schadt M., Leenhouts F., Appl. Phys. Lett. 50, 236 (1987)

[3] Scheffer T. J., Nehring J., J.Appl.Phys. 58, 3022 (1985)

[4] Sugiyama T., Kuniyasu S., Seo D., Jap. J. Appl. Phys. 29, 2045 (1990)

[5] Ogawa K., Mino N., Nakajima K., Jap. J. Appl. Phys. 29, L 1689 (1990)

[6] Blinov M., Kabayenkov A., Sonin A., Liq.Cryst. 5, 645 (1989)

[7] Rapini A., Papoular M., J. Phys. (Paris) 30, C4-54 (1969)

[8] Berreman D., J.Opt.Soc.Am 62, 502 (1972)

[9] Wöhler H., Haas G., Fritsch M., Mlynski D., J. Opt. Soc. Am. A5, 1554 (1988)

[10] Schmidt M., Schmiedel H., Mol. Cryst. Liq. Cryst., 172, 223 (1989)

[11] Keller P., Proc. SID 24, 317 (1983)

DYNAMICS OF THE FREEDERICKSZ TRANSITION IN NEMATIC LIQUID CRYSTALS

F. SAGUES
Departament Química Física
Universitat de Barcelona
Diagonal 647, BARCELONA 08028
Spain

Abstract- Within the general field of pattern formation studies, one of the most interesting situations corresponds to the analysis of a transient dynamical evolution to a stable steady pattern. Here we will deal with a particular realization of such a process which connects two homogeneous steady states through a transient inhomogeneous structure. The problem will be here addressed in relation with the Freedericksz transition in nematic liquid crystals. The general nematodynamic equations, incorporating hydrodynamic contributions and internal degrees of noise, will adopt the form of a Time Dependent Ginzburg Landau model. The pattern dynamics is monitored in terms of the temporal evolution of the structure factor for the orientational fluctuations.

I. INTRODUCTION

The magnetically induced Freedericksz transition occurs in a nematic slab when the director, describing the state of orientation of the nematic molecules inside the sample, reorientates following an applied magnetic field larger than a critical one H_c. The standard description corresponds to the appearance of distortions in the orientation with respect to the original one, the degree of distortion being homogeneous in each plane of the sample, and of maximum intensity in the mid plane far from the plates limiting the nematic material [1].

This simple picture of the Freedericksz instability is however too simplified to account for some experimental observed facts. Under appropriate conditions, transient spatial structures are found corresponding to modes of inhomogeneous distortions in the planes of the sample. A wide experimental evidence of this phenomenon, [2,3], supplemented with detailed theoretical analysis, [4,5], have been accumulating during these past years. The suggested explanation involves a dynamical coupling between the director field and the hydrodynamic motion associated with the reorientation. Such a coupling gives rise, during the transient process, to spatial domains with a well-defined periodicity.

The characteristic wavenumber of these transient patterns has been commonly described in terms of a most unstable mode. A linear analysis of the nematodynamic equations around the initial undistorted configuration identifies the mode of fastest growth. It is assumed that this mode dominates the transient dynamics. However, this approach although useful in understanding the main physical ingredients involved in the phenomenon, is far less valid if one is interested in the dynamics of the pattern formation process itself. With this specific aim we have recently proposed a model, [4], which permits us to identify the basic time scales governing the reorientation of the nematic sample.

The analysis is based on the evolution equation for the time-dependent structure factor which describes the orientational distortions of the director, once thermal fluctuations and hydrodynamic effects have been taken into consideration. A central role is played by an effective viscosity which accounts for a tradeoff of rotational for shear viscosities leading to a compromise at some intermediate nonzero wavenumber for which the increase in elastic energy contribution is favorably balanced by a higher energy dissipation rate In what follows we will describe this behavior for the simplest realizations of the magnetically induced Freedericksz transition.

II. PATTERN FORMATION EQUATIONS

The simplest geometry we can envisage corresponds to a twist geometry. In this situation the sample is contained between two plated perpendicular to the z axis. The director is initially aligned along the x axis, and the magnetic field is applied along the y axis. The transient behavior we will describe corresponds to the switch at t=0 from an initial value $H_i < H_c$ to a final one $H>H_c$. Under these conditions it has been experimentally shown that in addition to the usual twist deformations along the z axis, the system may transitorily develop a more complicated structure involving bend modes along the x direction (Fig.1).

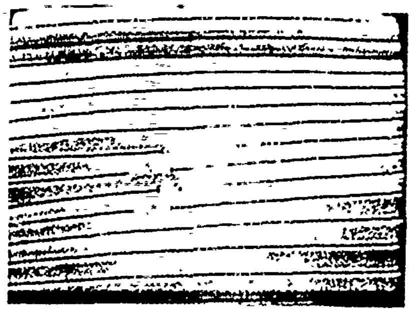


FIG.1 Photomicrograph of uniform periodic structure. Field strengh 7.4kG; temperature 25°C; sample thickness 50 μm; spacing between stripes ≈ 48 μm. (Reprinted from Y. W. Hui *et al.* J. Chem. Phys. 83, 288 (1985))

The appropriate scheme of nematodynamic equations is here used assuming for simplicity that macroscopic flow exists only along the y direction, and that homogeneity extends on the direction of the applied magnetic field. For typical on-plane reorientations we introduce the dinamical variable ϕ as

$$\begin{aligned} n_x(x,z) &= \cos\phi(x,z)\,, \\ n_y(x,z) &= \sin\phi(x,z)\,, \\ n_z &= 0\,. \end{aligned} \qquad (1)$$

A minimal coupling approximation is then invoked to obtain a pair of closed equations for ϕ and v_y:

$$d_t \begin{vmatrix} \phi \\ v_y \end{vmatrix} = \begin{vmatrix} -\frac{1}{\gamma_1} & \frac{1}{2\rho}(1+\lambda)\partial_x \\ \frac{1}{2\rho}(1+\lambda)\partial_x & \frac{1}{\rho^2}(\nu_2\partial_z^2+\nu_3\partial_x^2) \end{vmatrix} \begin{vmatrix} \frac{\delta F}{\delta\phi} \\ \frac{\delta F}{\delta v_y} \end{vmatrix} + \begin{vmatrix} \xi \\ \partial_x\Omega_{yx}+\partial_z\Omega_{yz} \end{vmatrix} \qquad (2)$$

The Gaussian random forces appearing in the above Langevin-type equations satisfy fluctuation-dissipation relations in terms of pure rotational and shear viscosities, respectively γ_1 and $\nu_{2,3}$. L and ρ stand respectively for a linear transversal dimension and the mass density of the sample. A series of technical manipulations are then strictly convenient. First, one introduces the hypothesis of negligible inertia which enables us to obtain a closed equation for the deformation angle. This equation is more easily handled in terms of a Fourier representation appropriate to the strong boundary conditions prescribed for the nematic at the limiting plates, z=±d/2:

$$\phi(x,z;t)=\sum_{m}\sum_{q_x}\theta_{m,q_x}(t)\cos(2m+1)\frac{\pi z}{d}e^{iq_x x},$$

$$\xi(x,z;t)=\sum_{m}\sum_{q_x}\xi_{m,q_x}(t)\cos(2m+1)\frac{\pi z}{d}e^{iq_x x}, \tag{3}$$

$$\Omega_{ya}(x,z,t)=\sum_{m}\sum_{q_x}\Omega^{a}_{m,q_x}(t)\cos(2m+1)\frac{\pi z}{d}e^{iq_x x}$$

The resulting equation for the amplitude of the reorientational mode is given by

$$\partial_t\theta_{m,q_x}(t)=\frac{1}{\bar{\gamma}_1}\left[\chi_a H^2-K_{22}(2m+1)^2\frac{\pi^2}{d^2}-K_{33}q_x^2\right]$$

$$\times\theta_{m,q_x}(t)+\eta_{m,q_x}(t), \tag{4}$$

$$Q=\frac{q_x}{(2m+1)\frac{\pi}{d}} \qquad \bar{\gamma}_1=\gamma_1-\frac{\alpha_2^2}{\eta_c+\eta_a Q^{-2}}$$

The important point to be noticed is that the temporal evolution of the reorientational process is no longer dictated by the pure rotational viscosity γ_1 but is governed by an effective wavenumber dependent viscosity $\bar{\gamma}_1$. Thus, the coupling of the director and velocity fields results in a reduction of the viscosity for all modes with $q_x \neq 0$. This permits modes of bend deformation along the x direction to grow faster than the homogeneous one, giving rise to pattern structuration. Actually, for $H > H_c = (K_{22}\pi^2/\chi_a d^2)$, K_{22} being the twist elastic constant and the anisotropic diamagnetic parameter, twist modes of wavenumber m become unstable. However due to the dependence of γ_1 on q_x, bend modes with $q_x = 0$ may lead the response of the system provided that

$$h^2(m)>1+\frac{K_{33}}{K_{22}}\frac{\bar{\eta}}{\bar{\alpha}}, \qquad h^2(m)=H^2/[(2m+1)^2H_c^2]$$

$$\alpha_2=-\tfrac{1}{2}\gamma_1(1+\lambda),$$

$$\bar{\alpha}=\alpha_2^2/\gamma_1\eta_c, \quad \bar{\eta}=\eta_a/\eta_c \qquad \eta_a=\nu_2, \tag{5}$$

$$\eta_c=\nu_3+\tfrac{1}{4}\gamma_1(1+\lambda)^2.$$

where K_{33} is the bend elastic constant and the remaining parameters in (5) are convenient and standard redefinitions of nematic viscosities. Thus, one predicts the appearance of a periodic pattern for magnetic fiels satisfying the above condition. This occurs for fields not much larger than the critical one $h^2 = 1$, although there still exists a range of magnetic intensities for which the homogeneous response dominates.

The early dynamical stages of the transition can be easily followed by converting Eq.(4) into an equation for the structure factor. Using standard methods one obtains

$$\partial_t C_{q_x,m}(t)=\frac{2}{\bar{\gamma}_1}\left[\chi_a H^2-K_{22}\left(\frac{(2m+1)\pi}{d}\right)^2-K_{33}q_x^2\right]$$

$$\times C_{q_x,m}(t)+\frac{2}{\bar{\gamma}_1}\frac{2k_BT}{V}. \tag{6}$$

A convenient way of monitoring the dynamical emergence of the pattern consists in analyzing the time evolution of the mode Q_{max} corresponding to the maximum of the structure factor for the most unstable twist deformation mode (m=0). This is depicted in Fig 2

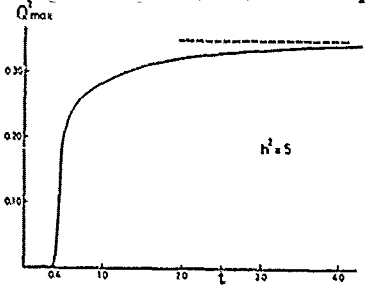


FIG.2 Maximum of the structure factor vs. time (taken adimensionalized in units of $\tau_0 = \gamma_1 / \chi_a H_c^2$). Parameter values concspond to those of MBBA (Ref.4).

Different and well-resolved time scales can be distinguished in this figure. A first time scale is associated to the sharp increase of Q_{max} when the system takes off from the initial conditions. This time should be interpreted as the characteristic time of appearance of the periodic pattern. A second time scale for the slow growth of Q_{max} is reasonably associated with the development of the inhomogeneous structure. Late stage dynamical scales accounting for the dissapearance of such transient patterns can not be described within the limits of this approach and instead, a proper analysis of the mobility and recombination of defect walls is more adequate [6]

Sometimes, the transient nature of the predicted periodicity is already apparent within the linear approximation here used. This is for example the case for slightly supercritical conditions in the splay geometry [7], as shown in Fig 3. This situation is relevant to recent experiments for the electrically induced Freedericksz instability [8]

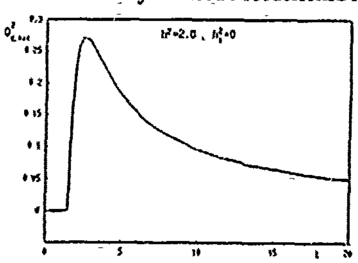


FIG.3 As in Fig. 2, but corresponding to the transversal component of the structure factor for the splay geometry (Ref.7).

REFERENCES

1. P.G. de Gennes, *The Physics of Liquid Crystals*, Clarendon, Oxford 1974
2. E.Guyon, R.Meyer and J.Salán, Mol.Cryst.Liq.Cryst. 54, 261 (1979)
3. F.Lonberg, S.Fraden, A.J.Hurd and R.B.Meyer, Phys.Rev.Lett.52,1903 (1984)
4. M.San Miguel and F.Sagués, Phys.Rev.A36,1883 (1987)
   and in *Pattern, Defects and Material Instabilities*, Eds. D.Walgraef and N.M.Ghoniem, Kluwer, Dordrecht 1990.
5. G.Srajer,S.Fraden and R.B.Meyer, Phys.Rev.A39,4828 (1989)
6. F.Sagués and M.San Miguel, Phys.Rev.A39,6567 (1989)
7. F.Arias and F.Sagués, preprint 1991
8. A.Buka, M.de la Torre Juarez, L.Kramer and I. Rehberg, Phys.RevA40,7427 (1989)

DIRECTOR PATTERN AND OPTICAL PERFORMANCE OF 2D INHOMOGENEOUS NEMATIC LC LAYERS

M. Schmidt, M. Grigutsch and H. Schmiedel
Universität Leipzig, Sektion Physik, Linnéstraße 5, O-7010 Leipzig

Abstract-We present a numerical computation of director configurations, switching behaviour and optical performance of a nematic liquid crystal (LC) layer in two dimensional inhomogeneous (2D) electrical fields. Coupling between director deformation and electrical field is fully included. The computations are applied to LC spatial light modulators (SLM). For the first time the anisotropy of the modulation transfer function of SLM is specified. Its dependence upon device and LC material parameters is studied.

I. INTRODUCTION

Usually, a nematic LC display is treated as a stratified medium. The corresponding one dimensional theory is well understood /1/. The optical performance of an LCD is computed in two steps: First, the distribution of the optical axis n in the LC layer (director pattern) is determined from continuum theory /3/. Second, the optical properties of the display are computed, commonly with Berreman's 4x4 matrix formalism /2/ for stratified media /1,5/.

In practice the optical response of the LC often depends on a spatially non-uniform electric field, resulting e. g. from pixel structures on matrix displays or intensity distributions on a photo conductor in LC SLMs /6/. With the ever decreasing display structures, director patterns in spatially inhomogeneous fields, limits of spatial and temporal resolution of LC layers and their optical performance become important.

II. CALCULATION OF DIRECTOR PATTERNS

We consider a nematic layer of thickness d between the planes z=0 and z=d of a Cartesian coordinate system under the influence of an electrostatic potential, with boundary conditions for the director tilt angle θ and the potential V:

$$\theta(x,z=0)=\theta_0 \qquad \theta(x,z=d)=\theta_0 \qquad (1)$$

$$V(x,z=0)=V_0 \qquad V(x,z=d)=V_1(x) \qquad (2)$$

We restrict ourselves to constant twist angle ϕ and two principal cases of director alignment: in the xz-plane of the field inhomogeneity (ϕ=0), and perpendicular to it (ϕ=90°). We assume invariance in y-direction.

Our aims are: 1) the determination of the equilibrium director pattern $\theta(x,z)$ and correspondirg electric potential V(x,z); and 2) the comptation of the director response to sudden changes in $V_1(x)$. The second problem can be treated as follows: Since the time constant of the electric field is much smaller than that of LC dynamics, we assume the electric field to be quasi-static and for a given director pattern $\theta(x,z,t)$ the corresponding potential V(x,z,t) is found from the equations of electrostatics of inhomogeneous media:

$$(\varepsilon_\perp+\Delta\varepsilon c^2\theta)V_{xx}+2\Delta\varepsilon s\theta c\theta(V_{xz}-V_x\theta_x+V_z\theta_z)+$$
$$+(\varepsilon_\perp+\Delta\varepsilon s^2\theta)V_{zz}+\Delta\varepsilon(c^2\theta-s^2\theta)(V_x\theta_z+V_z\theta_x)=0$$
$$\text{for } \phi=0 \quad (3a)$$

$$\varepsilon_\perp V_{xx}+(\varepsilon_\perp+\Delta\varepsilon s^2\theta)+2\Delta\varepsilon s\theta c\theta V_z\theta_z=0 \quad \text{for } \phi=90^\circ \quad (3b)$$

($V_{ij}=\partial^2V/\partial i\partial j$, $\theta_i=\partial\theta/\partial i$, $s\theta=\sin\theta$, $c\theta=\cos\theta$)

$\Delta\varepsilon=(\varepsilon_\parallel-\varepsilon_\perp)$, $\varepsilon_\parallel$ and $\varepsilon_\perp$ are the dielectric constants parallel and normal to n, resp. Now we integrate the nematodynamic equations 4a,b for $\theta(x,z,t)$ /3/ keeping V(x,z,t) constant during each integration step. Inertia and backflow are neglected.

$$\gamma_1\theta_t=\theta_{xx}(K_1s^2\theta+K_3c^2\theta)+\theta_{zz}(K_1c^2\theta+K_3s^2\theta)+$$
$$+(K_3-K_1)s\theta c\theta(2\theta_{xz}-\theta_x^2+\theta_z^2)+(K_3-K_1)(c^2\theta-s^2\theta)\theta_x\theta_z$$
$$+\varepsilon_o\Delta\varepsilon(s\theta c\theta(V_z^2-V_x^2)+(c^2\theta-s^2\theta)V_xV_z)$$
$$\text{for } \phi=0, \quad (4a)$$

$$\gamma_1\theta_t=K_2\theta_{xx}+\theta_{zz}(K_1c^2\theta+K_3s^2\theta)+$$
$$+(K_3-K_1)s\theta c\theta\theta_z^2+\varepsilon_o\Delta\varepsilon s\theta c\theta V_z^2 \quad \text{for } \phi=90^\circ.(4b)$$

The K_i are LC elastic constants, γ_1 is the rotational viscosity. We repeat this procedure to compute the complete director response using a finite difference method /4/. The resulting set of ordinary equations is solved by a relaxation method. An implicit time integration scheme is applied.

This procedure is applicable as well to the computation of the equilibrium director pattern. However, we prefer a faster approach. Eqs. (3,4) without the dissipative term $\gamma_1\theta_t$ are coupled self-consistently. They are solved by a multigrid relaxation method /4/ which speeds up computation remarkably.

III. OPTICAL PERFORMANCE OF A 2D-DEFORMED LIQUID CRYSTAL LAYER

Now we calculate the optical performance of the untwisted nematic layer with a given director tilt profile $\theta(x,z)$. For normal light incidence and a director tilt profile varying in x-direction slowly compared to the wavelength λ of the light we can use the geometrical optics approximation (GOA) /7/ and describe the light propagation inside the LC layer as locally one-dimensional. The transmitted intensity is

$$T(x)=\sin^2\left(\frac{\pi d}{\lambda}(\bar{n}_{e,eff}(x)-n_o)\right) \qquad (5)$$

with n_o, n_e and $\bar{n}_{e,eff}(x)$ being the ordinary, extraordinary and effective extraordinary refractive indices of the nematic.

IV. OPTICAL PERFORMANCE AND RESOLUTION OF SLM

We consider a transmissive SLM consisting essentially of LC layer and photoreceptor. Resolution and field fringing of the photoreceptor have been investigated recently /6,8/, the influence of the LC layer was however neglected. In this paper we study the resolution capability of the LC layer and calculate the optical response to harmonic boundary voltage distributions

$$V_1(x) = \overline{V}_0 + \frac{1}{2}\,\overline{V}_1(t)\left(1-\cos\left[\frac{2\pi x}{g}\right]\right) \qquad (6)$$

which are caused by periodic intensity modulated stripe patterns imaged at the photoreceptor. In the following we set the potential at the second electrode $V_0=0$. For given material and device parameters one has to compute now the electro-optic characteristics between crossed polarizers from the one-dimensional theory. Optimum bias $\overline{V}_0$ and amplitude $\overline{V}_1$ are determined by the requirement of maximum contrast and minimum response time. In the computation we used the parameters $K_1=11.3$pN, $K_2=8$pN, $K_3=12.5$pN, $\varepsilon_\parallel=24.8$, $\varepsilon_\perp=6.4$, $\theta_0=0$, $n_e=1.610$, $n_o=1.489$, $\gamma_1=78.3$mPas, $\lambda=550$nm, d=0.010mm. We compare two possible switching regions: switching between the first two extrema of the transmission vs. voltage curve ($\overline{V}_0=0.94$V, $\overline{V}_1=0.27$V), and between the last two extrema ($\overline{V}_0=1.51$V, $\overline{V}_1=0.76$V). Figure 1 shows the transmitted intensity as a function of x for $\phi=0$ and $\phi=90°$ and both switching regions calculated according to eq. (5) with an assumed grating constant g=0.020mm.

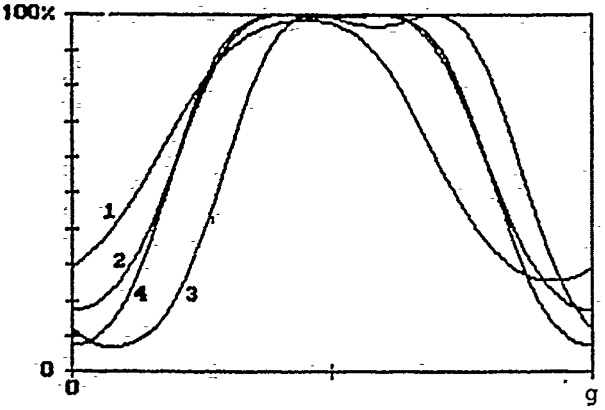


Fig. 1. Light transmittance as function of x. $\phi=0$ for curves 1,3, and $\phi=90°$ for 2,4. Switching between the first two extrema (1,2) and the last two extrema (3,4).

One can see, that imaging is always better for $\phi=90°$ compared to $\phi=0$. The $\phi=0$ transmission curve becomes strongly asymmetric in the high voltage region. This was already expected from eqs. (3a),(4a) whereas eqs. (3b),(4b) are invariant with respect to a transformation x → -x.

Usually, the modulation transfer function

$$MFT = (T_{max}-T_{min})/(T_{max}+T_{min}) \qquad (7)$$

is introduced to characterize the resolution capability of the layer. T_{min} and T_{max} are the minimum and maximum transmittances of the read beam. Fig. 3 shows the MFT vs. grating constant g. The contrast decreases rapidly with decreasing g and reaches a saturation at large g. The calculations show that the resolution of an SLM improves with decreasing layer thickness d and large $\Delta\varepsilon/\varepsilon_\perp$. It is optimum at $K_3/K_1 \cong 1$. For $\phi=90°$ the resolution is higher with decreasing K_2. Resolution can be further improved by raising the surface pretilt angle up to $\theta_0=10°$.

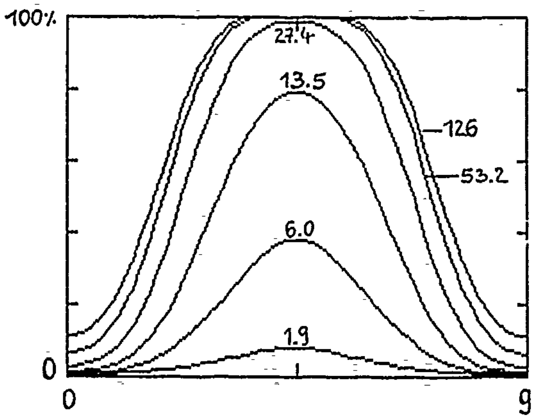


Fig. 2. Time development of transmittance for $\phi=90°$ when $\overline{V}_1$ is switched from 0 to 0.76V in the high voltage switching region (cf. curve 4 of fig. 1). The parameter gives the time in ms after switching $\overline{V}_1$.

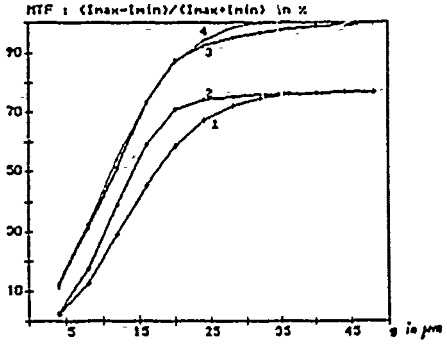


Fig.3: MTF as a function of the grating g for the 0.01mm cell (notation of the graphs corresponds to that of fig.1).

REFERENCES

1. K. Eidner, G. Mayer, M. Schmidt, and H. Schmiedel, *Mol. Cryst. Liq. Cryst.* 172, 191 (1989) and references therein.
2. D. W. Berreman, *J. Opt. Soc. Am.* 62, 502 (1972).
3. P. G. de Gennes, *The Physics of Liquid Crystals*, Clarendon Press, Oxford 1974.
4. G. D. Smith, *Numerische Lösung von partiellen Differentialgleichungen*, Akademie-Verlag Berlin 1971.
5. H. Wöhler, G. Haas, M. Fritsch, and D. A. Mlynski, *J. Opt. Soc. Am.* A5, 1554 (1988).
6. D. Armitage, J. I. Thackara, and W. D. Eades, *Appl. Optics* 28, 4763 (1989).
7. H. L. Ong, R. B. Meyer, *J. Opt. Soc. Am.* A2, 198 (1985).
8. D. Armitage, J. I. Thackara, and W. D. Eades, *Proc. SPIE* 936, 56 (1988).

THE OPTIMISATION AND ANALYSIS OF LCD PERFORMANCE USING NUMERICAL MODELLING

M C K WILTSHIRE
GEC-Marconi Limited
Hirst Research Centre
Wembley, Middlesex, HA9 7PP
United Kingdom

Abstract - Numerical modelling has been used to optimise the performance of liquid crystal displays. The modelling and optimisation are described, and the predicted performance is compared with that of actual displays. The accuracy of the modelling permits its use in LCD fault diagnosis and as a research tool in exploring novel liquid crystal systems.

I. INTRODUCTION

Liquid crystal displays (LCDs) are being used in an ever-increasing variety of applications. Although the bulk of the LCD market is still in simple watch and calculator displays, their use as complex alphagraphic panels for lap-top computers, portable televisions, car and cockpit instrumentation, and in-flight entertainment systems, is growing rapidly. These high added-value products need to be well-tailored to the customer's requirements, so that the ability to optimise their performance has become increasingly important. Whereas the simple watch displays can be designed analytically or by rule-of-thumb, complex displays and those requiring good off-axis performance must be optimised using a numerical model. This paper describes the methods adopted at GEC for this purpose and presents some examples of our results, comparing the predicted performance with that achieved in practice.

II. NUMERICAL MODEL

The modelling problem falls into two parts. First, the LC director profiles, i.e. $\underline{n}(z)$, must be calculated for the different display states and, second, the optical properties of these must be calculated. We follow the methods developed by Berreman [1], and find the $\underline{n}(z)$ that both satisfies the boundary conditions imposed by the cell construction and the applied voltage, and is a minimum of the total Helmholtz free energy F where

$$2F = K_{11}(div\,\underline{n})^2 + K_{22}(\underline{n}.curl\,\underline{n} - 2\pi/P)^2 + K_{33}(\underline{n}\times curl\,\underline{n})^2 + \underline{D}.\underline{E} \quad (1)$$

Here the K_{ii} are the elastic constants and P the cholesteric pitch of the LC fluid. The device is assumed uniform in the plane of the LC so that $\underline{n}$ and $\underline{D}$ are functions of z alone. Then $D(z) = \epsilon(z)\epsilon_0 E(z) =$ constant, and $\epsilon(z)$ is calculated from $n(z)$ and the (uniaxial) dielectric tensor with elements $\epsilon_{\parallel}$ and $\epsilon_{\perp}$. Following Berreman [1], we adopted a shooting method which is interactive, but automatically alerts the user to conditions where the profile is changing rapidly and avoids metastable configurations.

The optical properties of these configurations are calculated using the 4x4 matrix method developed to treat stratified anisotropic planar structures [2]. The field amplitudes at the two surfaces of the structure are related by a propagation matrix $\underline{\underline{P}}$ so that

$$\underline{\psi}(z) = \underline{\underline{P}}\,\underline{\psi}(0) \quad (2)$$

where $\underline{\psi} = (E_x, H_y, E_y, -H_x)^T$. $\underline{\underline{P}}(z)$ is obtained by writing a differential form of (2) i.e.

$$\underline{\psi}'(z) = i\nu\underline{\underline{\Delta}}(z)\underline{\psi} = \underline{\underline{D}}(z)\underline{\psi} \quad (3)$$

in each stratum of the medium, within which the optical properties, contained in $\underline{\underline{\Delta}}(z)$, are constant. Then

$$\underline{\underline{P}}(z) = \exp\left\{\int_0^z \underline{\underline{D}}(z)dz\right\} \quad (4)$$

permits the calculation of $\underline{\underline{P}}$ from the dielectric tensor elements and orientation as a function of z. Having calculated $\underline{\underline{P}}$ it is then straightforward to obtain the transmission and reflection coefficients.

This calculation must be repeated for each wavelength of interest across the visible spectrum, 380-800 nm, and as a function of the angles of the polarisers and any other layers in the device. It must therefore be done efficiently. From (2), $\underline{\underline{P}}_{tot}$ for the entire structure is the product of the $\underline{\underline{P}}$'s for each component, with the appropriate orientational factors. For an LCD with polarisers set at (θ_1, θ_2)

$$\underline{\underline{P}}_{tot} = \underline{\underline{R}}^{-1}(\theta_2)\underline{\underline{P}}_{POL}\underline{\underline{R}}(\theta_2).\underline{\underline{P}}_{LC}.\underline{\underline{R}}^{-1}(\theta_1)\underline{\underline{P}}_{POL}\underline{\underline{R}}(\theta_1) \quad (5)$$

where $\underline{\underline{P}}_{POL}$ and $\underline{\underline{P}}_{LC}$ describe the polariser and LC layers respectively and $\underline{\underline{R}}(\theta)$ is a rotation matrix. Thus the problem of optimising a display is reduced to performing the matrix multiplication (5) on previously calculated $\underline{\underline{P}}$ matrices.

III. DISPLAY OPTIMISATION

The display optimisation process starts by calculating the director profiles in the OFF- and ON-states of the display. For this, we require the elastic constants and permittivities of the LC fluid, the physical parameters of the cell (i.e. thickness, twist and surface tilt), and the drive voltages V_{NS} and V_S. These voltages are related by the level of multiplexing N through

$$V_S/V_{NS} = M = [(\sqrt{N}+1)/(\sqrt{N}-1)]^{1/2},$$

and are equally disposed about the central switching voltage V_c for which the mid-plane tilt = 45°.

From these profiles and the refractive indices of the LC, the $\underline{\underline{P}}$ matrices for the LC layer in the two states are calculated and stored for 40 different wavelengths across tne visible spectrum. It is essential to include the dispersion of the LC. The $\underline{\underline{P}}$-matrix of the polariser is also calculated as a function of wavelength. Then equation (5) is used to calculate P_{tot} and hence the transmission spectra for the range of polariser angles of interest. The transmission spectra, $T(\lambda)$, are combined with an illuminant spectrum $S(\lambda)$ to give the CIE tristimulus co-ordinates X, Y, Z where

$$X = k\int_{370}^{770} S(\lambda)T(\lambda)\bar{x}(\lambda)d\lambda \quad etc. \quad (6)$$

Here $\bar{x}(\lambda)$ is the colour matching function and k is a normalising factor

$$k = 100/\int_{370}^{770} S(\lambda)\bar{y}(\lambda)d\lambda \quad .$$

Then the tristimulus Y value is the luminance transmission or "brightness" of the display and Y_{OFF}/Y_{ON} is its contrast ratio. Contour plots of iso-contrast and brightness versus the polar angles enable the optimum display configuration to be selected.

In many cases, a trade-off must be made between contrast and brightness. For the case shown in Fig.1, the polariser angles were selected to provide a contrast ratio >20 with reasonable ON-state transmission when a narrow band green illuminant was used. The predicted transmission spectrum is compared with that measured for the actual display in Fig.1.

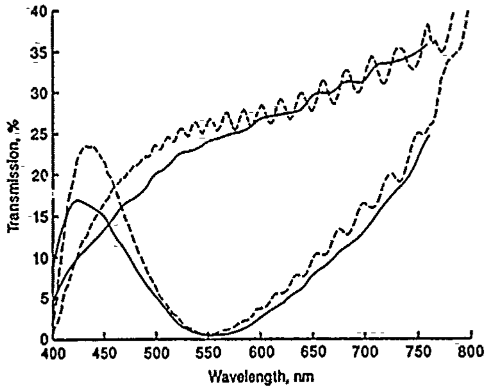


Fig.1 Comparison of calculated (solid line) and measured (broken line) transmission spectra for an optimised SBE display.

IV BLACK AND WHITE SUPERTWIST OPTIMISATION

Supertwist LCDs are inherently coloured, which limits their applicability. The colouration can be removed by including a birefringent film to compensate the optical properties of the LCD. This compensation cannot be ideal, so optimisation is necessary. The method is an extension of equation (5). We write:

$$\underline{\underline{P}}_{tot} = \underline{\underline{R}}^{-1}(\theta_2)\underline{\underline{P}}_{pol}\underline{\underline{R}}(\theta_2).\underline{\underline{R}}^{-1}(\theta_3)\underline{\underline{P}}_{B}\underline{\underline{R}}(\theta_2).\underline{\underline{P}}_{LC}.\underline{\underline{R}}^{-1}(\theta_1)\underline{\underline{P}}_{pol}\underline{\underline{R}}(\theta_1). \quad (7)$$

where θ_3 is the orientation of the birefringent layer and $\underline{\underline{P}}_B$ is its $\underline{\underline{P}}$-matrix. This can be calculated analytically from its retardation which, in turn, depends on the film thickness. Hence there are now four variables against which to optimise the display: two polariser angles θ_1, θ_2, the film axis angle θ_3 and its thickness. The optimisation criterion was taken as the blackest black state along with the whitest white state, so we numerically minimised

$$\Delta E^2 = (L^{*2} + u^{*2} + v^{*2})_{dark} + \left((L_o^* - L^*)^2 + u^{*2} + v^{*2}\right)_{light} \quad (8)$$

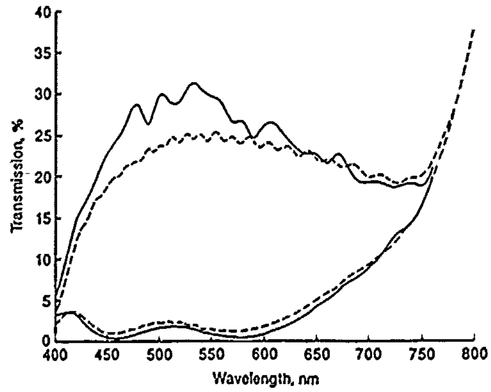


Fig.2 Comparison of calculated (solid line) with measured (broken line) transmission spectra of a film compensated black and white SBE display.

where L^*, u^*, v^* are the display's co-ordinates in the CIE 1973 uniform colour space and L_o^* is the brightness of a pair of parallel polarisers. The optimum values for the parameters were used to construct a display. Fig.2 shows a comparison of the calculated and measured spectra.

V DISCUSSION

The numerical model has proved extremely accurate for LCD design. Besides the examples quoted above, it has been possible to design displays with minimal colour shift between their (reflective) daylight operation and the (transmissive) backlit mode even though the backlight was a blue-green electroluminescent panel. We have also designed displays for NVG compatibility without using extra filters. It should also be emphasised that the modelling is not confined to normal incidence: Fig.3 shows a calculated iso-contrast plot for an SBE display optimised for two observers at ±40°.

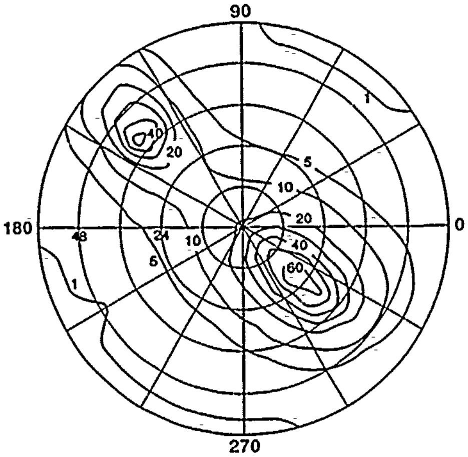


Fig.3 Calculated iso-contrast plot for an SBE display optimised for viewing at ±40° incidence. The display is mounted so that the rear LC alignment direction ($\phi = 0$) is at 45°

The modelling programs are sufficiently accurate that realistic simulations, both static and dynamic, of LCD appearance have been implemented [3].

Besides designing displays, the modelling programs can be used to investigate faults in displays. For example, SBE displays are very sensitive to the precise orientation of their polarisers. It is impossible to measure these angles after cell assembly, but by measuring the display performance (i.e. its contrast ratio and transmission) the polariser orientation can be accurately deduced from a comparison with the model predictions. Thus faults in the display fabrication can be readily identified.

Finally, it should be noted that these programs are an invaluable research tool. In ferroelectric LCDs, the director profile is generally unknown. By measuring their optical properties and comparing these with the predicted characteristics of plausible models, a predictive capability for FLCDs can also be developed. In conclusion, the numerical modelling programs for LCDs have proved to be very accurate. They have been used to design custom displays, diagnose faults and as a tool in display research.

REFERENCES

1 Berreman D W, Phil. Trans. Roy. Soc. A309 203-216 (1983)
2 Berreman D W, J. Opt. Soc. Am. 62 502-10 (1972), ibid 63 1374-80 (1973)
3 Placencia Porrero I and van der Meulen A E, Proc. Eurodisplay '90, 106-10 (1990)

CONTINUUM THEORY OF SMECTIC C LIQUID CRYSTALS; STATIC CONFIGURATIONS AND ELEMENTARY FREDERIKS TRANSITIONS

I. W. STEWART
Department of Theoretical Mechanics,
University of Nottingham,
University Park,
Nottingham, NG7 2RD, U.K.

and

F.M. LESLIE and T. CARLSSON
Mathematics Department, Livingstone Tower,
Strathclyde University,
26 Richmond Street,
Glasgow, G1 1XH, U.K.

ABSTRACT - Smectic C liquid crystals are shown to exhibit six types of static configuration which satisfy the equilibrium equations of the continuum theory introduced in [1]. A brief mathematical description of the resulting families of parallel surfaces which correspond to these static solutions is given. A possible configuration is a wedge of concentric cylinders, a Frederiks transition induced by an appropriate electric field is discussed in this case.

I. INTRODUCTION

To describe the layer structure of a smectic C liquid crystal we introduce the unit layer normal $\mathbf{a}$ which is subject to the constraint [2]

$$\nabla \times \mathbf{a} = \mathbf{0} \ . \tag{1}$$

As in [3], a unit vector $\mathbf{c}$ which is perpendicular to $\mathbf{a}$ is used to describe the direction of tilt of the alignment with respect to the layer normal. The two directors $\mathbf{a}$ and $\mathbf{c}$ satisfy the constraints

$$\mathbf{a}.\mathbf{a} = \mathbf{c}.\mathbf{c} = 1 \ , \quad \mathbf{a}.\mathbf{c} = 0 \ . \tag{2}$$

The energy integrand used in smectic C theory is

$$\begin{aligned} 2W = {} & K_1(\nabla.\mathbf{a})^2 + K_2(\nabla.\mathbf{c})^2 + K_3(\mathbf{a}.\nabla\times\mathbf{c})^2 + K_4(\mathbf{c}.\nabla\times\mathbf{c})^2 \\ & + K_5(\mathbf{b}.\nabla\times\mathbf{c})^2 + 2K_6(\nabla.a)(\mathbf{b}.\nabla\times\mathbf{c}) + 2K_7(\mathbf{a}.\nabla\times\mathbf{c})(\mathbf{c}.\nabla\times\mathbf{c}) \\ & + 2K_8(\nabla.\mathbf{c})(\mathbf{b}.\nabla\times\mathbf{c}) + 2K_9(\nabla.\mathbf{a})(\nabla.\mathbf{c}) \ , \end{aligned} \tag{3}$$

where the K_i are elastic constants (see [1],[4],[5]). The corresponding equilibrium equations examined are (see [1],[6])

$$\left.\begin{aligned} & \left(\frac{\partial W}{\partial a_{i,j}}\right)_{,j} - \frac{\partial W}{\partial a_i} + G_i^a + \lambda a_i + \mu c_i + e_{ijk}\beta_{k,j} = 0 \\ & \left(\frac{\partial W}{\partial c_{i,j}}\right)_{,j} - \frac{\partial W}{\partial c_i} + G_i^c + \kappa c_i + \mu a_i = 0 \end{aligned}\right\} \tag{4}$$

where β, λ, μ and κ are Lagrange multipliers arising from the constraints (1) and (2) and $\mathbf{e}$ is the usual alternator. Solutions to (4) provide static configurations for smectic C liquid crystals.

II. STATIC SOLUTIONS

There are six types of well behaved surface which readily provide static solutions for a restricted six term version (K_1-K_6) of the energy given in (3), namely, the Dupin cyclides, circular tori of revolution, spheres, parabolic cyclcides, infinite cylinders and planes (see [1],[5]-[7]). Solutions to equations (4) for $\mathbf{a}$ and $\mathbf{c}$, including any related Lagrange multipliers, can be found and as an example we now mention the parabolic cylcides [7].

The Cartesian equation of a parabolic cyclide is [7]

$$x(x^2+y^2+z^2)+(x^2+y^2)(\ell-\mu)-z^2(\ell+\mu)-(x-\mu+\ell)(\ell+\mu)^2 = 0 \tag{5}$$

where the confocal parabolae essential to its construction are

$$\left.\begin{aligned} y^2 &= 4\ell(x+\ell) \\ z &= 0 \end{aligned}\right\} \qquad \left.\begin{aligned} z^2 &= -4\ell x \\ y &= 0 \end{aligned}\right\} \tag{6}$$

and ℓ and μ are real parameters. Equation (5) may be parameterized as

$$\begin{aligned} x(1+\theta^2+t^2) &= \mu(\theta^2+t^2-1) + \ell(t^2-\theta^2-1) \\ y(1+\theta^2+t^2) &= 2t(\ell(\theta^2+1)+\mu) \\ z(1+\vartheta^2+t^2) &= 2\theta(\ell t^2-\mu) \end{aligned} \tag{7}$$

where $-\infty < \mu,\theta,t < +\infty$ (varying θ and t while keeping μ fixed gives one particular parabolic cyclide surface). Transforming from the Cartesian to the (μ,θ,t) frame we can show that

$$\mathbf{a} = (1,0,0) \quad \text{and} \quad \mathbf{c} = (0,1,0) \tag{8}$$

satisfy the constrains (1) and (2) and solve the transformed version of (4) upon a suitable choice of Lagrange multipliers. The consequences of varying ℓ and μ are discussed in [7].

III. FREDERIKS TRANSITION IN A WEDGE

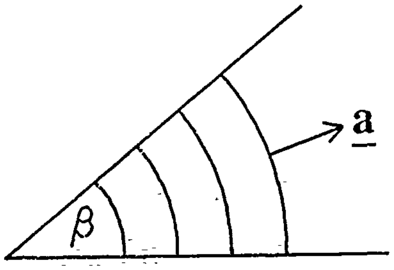


Fig. 1 The smectic layers are assumed to be parts of concentric cylinders with the common axis coinciding with the centre of the wedge. The layer normal $\mathbf{a}$ is in the $\hat{\mathbf{r}}$ direction and the angle between the boundary plates is β.

Introducing the vector

$$\mathbf{b} = \mathbf{a} \times \mathbf{c} \tag{9}$$

we can rewrite the energy integrand as (see [4],[8])

$$\begin{aligned} 2W = {} & A_{12}(\mathbf{b}.\nabla\times\mathbf{c})^2 + A_{21}(\mathbf{c}.\nabla\times\mathbf{b})^2 + 2A_{11}(\mathbf{b}.\nabla\times\mathbf{c})(\mathbf{c}.\nabla\times\mathbf{b}) \\ & + B_1(\nabla.\mathbf{b})^2 + B_2(\nabla.\mathbf{c})^2 + B_3[\tfrac{1}{2}(\mathbf{b}.\nabla\times\mathbf{b}+\mathbf{c}.\nabla\times\mathbf{c})]^2 \\ & + 2B_{13}(\nabla.\mathbf{b})(\mathbf{b}.\nabla\times\mathbf{b}+\mathbf{c}.\nabla\times\mathbf{c}) + 2C_1(\nabla.\mathbf{c})(\mathbf{b}.\nabla\times\mathbf{c}) \\ & + 2C_2(\nabla.\mathbf{c})(\mathbf{c}.\nabla\times\mathbf{b}) \end{aligned} \tag{10}$$

where the A_i, B_i and C_i are elastic constants. Introducing the cylindrical coordinate system (r,α,z) we set

$$\begin{aligned} \mathbf{a} &= \hat{\mathbf{r}} \\ \mathbf{b} &= -\hat{\boldsymbol{\alpha}} \cos\varphi + \hat{\mathbf{z}} \sin\varphi \\ \mathbf{c} &= \hat{\boldsymbol{\alpha}} \sin\varphi + \hat{\mathbf{z}} \cos\varphi \end{aligned} \tag{11}$$

and assume that $\varphi = \varphi(\alpha)$ with $\varphi(0) = \varphi(\beta) = 0$ where β is the wedge angle (see Fig. 1). To induce a Frederiks transition we apply an electric field of the form

$$\mathbf{E} = \frac{U}{r\beta}\,\hat{\boldsymbol{\alpha}} \tag{12}$$

across the bounding plates of the wedge of concentric cylinders where U is the voltage. The corresponding electrical energy integrand is

$$2W_e = -\varepsilon_a\varepsilon_0\left[\frac{U}{r\beta}\right]^2 \sin^2\theta \, \sin^2\varphi \tag{13}$$

where ε_a is the dielectric anisotropy of the liquid crystal, ε_0 is the permittivity of free space and θ is the usual fixed tilt angle associated with smectic C liquid crystals. To obtain a threshold for a Fredericks transition we add the integrands (10) and (13) and minimize the integral of this sum with respect to α over the region $r_1 < r < r_2$, $0 < \alpha < \beta$, $z_1 < z < z_2$. This is achieved by solving the following linearized Euler–Lagrange equation (see [8])

$$\bar{B}_2 \frac{d^2\varphi}{d\alpha^2} + 2(\bar{A}_{21}+\bar{A}_{11})\varphi + \varepsilon_a\varepsilon_0\left[\frac{U}{\beta}\right]^2\varphi = 0 \qquad (14)$$

where $\bar{B}_2$, $\bar{A}_{21}$ and $\bar{A}_{11}$ are the suitably adjusted temperature independent parts of the corresponding original elastic constants appearing in (10). To satisfy $\varphi = 0$ at $\alpha = 0,\beta$ we make the ansatz

$$\varphi(\alpha) = \varphi_m \sin\left[\frac{\pi}{\beta}\alpha\right] \qquad (15)$$

where φ_m is suitably small. Inserting this ansatz into equation (14) we derive the Frederiks theshold U_c as

$$\varepsilon_a\varepsilon_0 U_c^2 = \pi^2\bar{B}^2 - 2\beta^2(\bar{A}_{21}+\bar{A}_{11}).$$

We therefore see that by varying the wedge angle β, we can evaluate the constants $\bar{B}_2$ and $(\bar{A}_{21}+\bar{A}_{11})$ by measuring the Fredericks threshold for each value of β.

REFERENCES

[1] "Smectic C Liquid Crystal Continuum Theory", F.M. Leslie, M Nakagawa and I.W. Stewart, *to appear.*

[2] C.W. Oseen, Trans. Faraday Soc., 29, 883, (1933).

[3] P.G. de Gennes, "The Physics of Liquid Crystals", O.U.P., 1974.

[4] "Equivalent Smectic C Liquid Crystal Energies", F.M. Leslie, I.W. Stewart, T. Carlsson and M. Nakagawa, *to appear.*

[5] "A Theoretical Study of Smectic Focal Domains", M. Nakagawa, J. Phys. Soc. (Japan), 59, 81 (1990).

[6] "A Continuum Theory for Smectic C Liquid Crystals", F.M. Leslie, I.W. Stewart and M. Nakagawa, *to appear in* Mol. Cryst. Liq. Cryst. (1991).

[7] "Smectic Liquid Crystals and the Parabolic Cyclides", I.W. Stewart, M. Nakagawa and F.M. Leslie, *to appear.*

[8] "Theoretical Studies of Smectic C Liquid Crystals Confined in a Wedge", T. Carlsson, I.W. Stewart and F.M. Leslie, *to appear in* Liquid Crystals (1991).

OPTICAL TRANSMISSION MODELLING OF FERROELECTRIC LCDs[1]

J.M. Otón, F. Olarte, F.J. López-Hernández and F. Muñoz-Latrás

Dept. Tecnología Fotónica, ETSI Telecomunicación, Ciudad Universitaria
28040 Madrid Spain

Abstract.- Modelling of ferroelectric liquid crystal displays involves angular and spectral transmission of real backlights, using commercially available polarisers and actual FLC mixtures. Running times for a complete case may be extremely large. Therefore optimised procedures and simplified models for precalculations become necessary. A fairly accurate description of the optical behavior of real FLC displays in working conditions has been achieved. Polariser angles optimisation, spectral and angular transmission plots and color coordinates calculations are the most useful results.

I. INTRODUCTION

The use of ferroelectric liquid crystals (FLC) for manufacturing high quality displays has drawn considerable attention since surface stabilized (SSFLC) bistable structures of these materials were reported by Clark and Lagerwall [1]. A simple bookshelf geometry with uniform layers perpendicular to the glass plates was proposed, and a fast electrooptical switching was demonstrated. Further work by them and other researchers [2-4], however, showed that the usual configurations found in real structures are far more complicated. tilted bookshelf, splayed or chevron.

Optical transmission models may greatly help to develop these devices. Under the point of view of commercial displays, the most interesting information that a computer model may provide is the one related to optical transmission properties of actual devices in working conditions.

II. PROBLEM FORMULATION

Two models for angular and spectral distribution of light transmitted through FLC samples have been prepared and tested. The first model -called slab model- can compute light transmission through FLC cells whose director orientations are defined by a small number of homogeneous slabs. The mathematical description of the optical system (i.e., polariser, glass, anisotropic material, glass, and polariser) is based on geometrical optics [5].

The optical transmission calculations are performed for all possible angles (0-360°, 10° steps) and oblique incidences (0-80°, 5° steps). Driven and zero volts states are computed by suitably orienting the liquid crystal slab directors within the optical system. Data for real polarisers and backlights (380-770 nm, every 10 nm) are then included, and the overall transmission for every angle is computed.

The second, so-called continuous-varying profile (CVP) model, is based on Berreman 4x4 formulation [6,7] and may be used for any arbitrary director profile. Input data and output results are defined as above. The truncated series expansion of Berreman's method is avoided using explicit expressions for 4x4 propagation matrices of homogeneous slices of the material, as proposed by Wöhler et al. [8]. Results from both methods are similar, the former being faster but restricted to simplest profiles.

III. RESULTS

In the examples given below, the following elements are used:
Backlight: C.I.E. D65
Color filters. EMI RGB/standard fluorescent lamp
Polarizers: G1220
Material: ZLI 3654 (Merck).

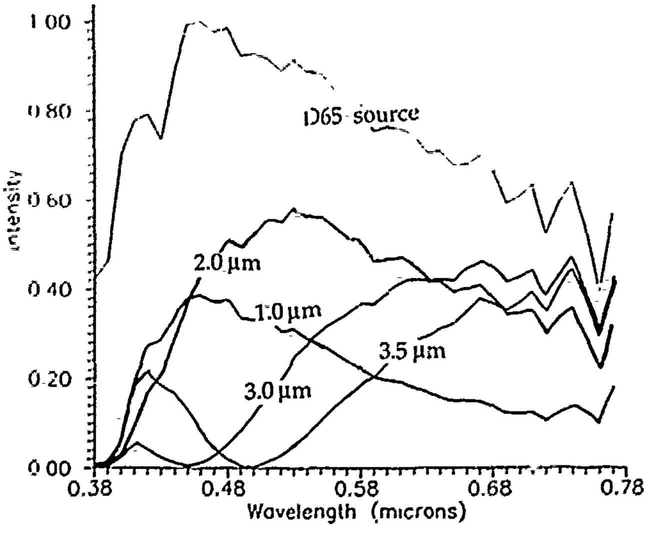
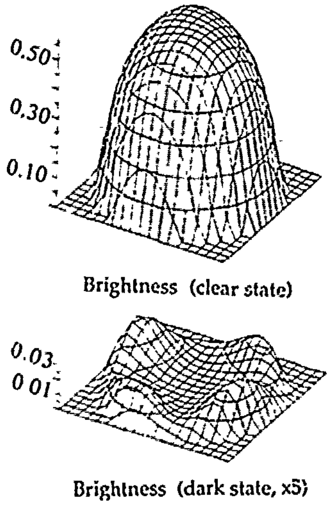


Figure 1.- Transmission spectra for different thicknesses. Backlight: D65. Slab model.

[1] This work is supported in part by ESPRIT II Project 2360 FELICITA and TIC Program from spanish CICyT.

The high birefringence of the clear state in FLC displays may influence the output of wide spectrum backlights. Figure 1 shows the spectral response of a 15° tilted bookshelf with a 3° surface pretilt angle. The region of shorter wavelengths is remarkable affected for higher thicknesses. The clear state is behaving like an almost ideal retardation plate. The models predict little spectral changes for samples below 2µm for a typical birefringence. This should be considered a design parameter, specially in colour displays, where luminance of the blue filter may be considerably reduced.

Figure 2 is a 3D plot of oblique incidences. A 15° chevron with 10° pretilt at 30 °C is represented. The central region of the plot corresponds to nearly normal incidences; angles are linearly varying with distance from the centre. The outer regions are grazing incidences up to 80°. Rotation of the viewing angle is achieved by rotating the figure. While the clear state yields a fairly symmetric hat shape, the dark state shows small maxima at 45° for nearly grazing incidence. This is due to residual light passing through non-ideal polarisers. Contrast curves may be directly derived from these plots.



Brightness (clear state)

Brightness (dark state, x5)

Figure 2.- 3D plots normalised to source luminance through parallel ideal polarisers. CVP model.

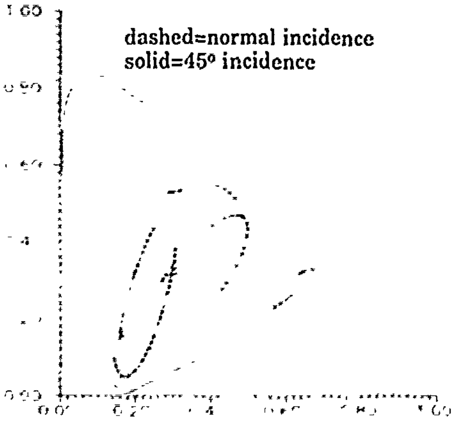


Figure 3.- Evolution of x, y C.I.E. colour coordinates with sample thickness (0-5µm). D65 and RGB filters are represented. CVP model.

Transmission data may be transformed into C.I.E. colour coordinates. Figure 3 shows the evolution of dominant colours and saturation when sample thickness is varied from 0 to 5 µm. No other cell parameter (tilt angle, pretilt, birefringence, FLC configuration, oblique incidence, chevron/tilted bookshelf angle) produces any relevant colour migration. However, sample thickness (strictly, the product Δn.d) produces dramatic effects in colour coordinates as it does in luminance (Fig. 1). Again thicknesses below 2µm are advisable. The central curve of figure 3 shows a D65. The outer curves are red, green and blue filters currently used for commercial colour displays.

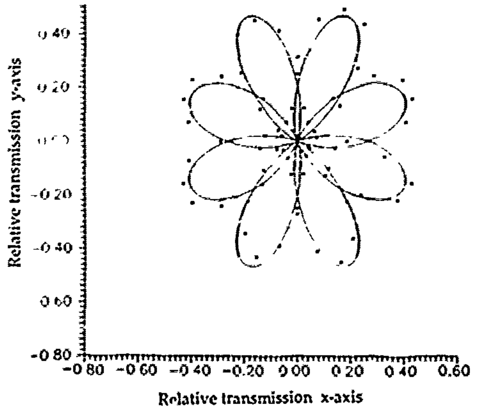


Figure 4.- Experimental (*) and predicted (solid) 45° oblique transmission of a 2µm FLC cell rotating between fixed crossed polarisers. Slab model.

An experimental setup consisting of a He-Ne laser, two fixed crossed polarisers and a FLC sandwich twisted 45° has been prepared to test oblique incidences. The sample is rotated 360° while switching. The output in polar coordinates (Fig. 4) is a daisy-shaped curve where four "petals" are maxima from one of the states interlaced with another four maxima from the other state. Both theoretical and experimental data are normalised using normal incidence. A fair match of the results is found, specially in the angular values of the maxima. These values are related to the projection of the tilt angle on the plane of incidence thus giving information of the actual FLC configuration.

REFERENCES

[1] N.A. Clark and S.T. Lagerwall *Appl. Phys. Lett.* 36 (11) 899 (1980).

[2] N.A. Clark and S.T. Lagerwall *Proc. 6th Int. Display Res. Conf.*, Tokyo, Japan (1986) p. 456.

[3] T.R. Ricker, N.A. Clark, G.S. Smith, D.S. Parmar, E.B. Sirota and C.R. Safinya *Phys. Rev. Lett.* 59 2658 (1987).

[4] Y. Ouchi, Ji Lee, H. Takezoe and A. Fukuda *Jpn. J. Appl. Phys. 2, Lett.* 27 (5) 725 (1988).

[5] H.L. Ong *J. Appl. Phys.* 64 614 (1988).

[6] D.W. Berreman *J. Opt. Soc. Am.* 62 (4) 502 (1972).

[7] D.W. Berreman *J. Opt. Soc. Am.* 63 (11) 1374 (1973).

[8] H. Wöhler, G. Haas, M. Fritsch and D.A. Mlynski *J. Opt. Soc. Am. A* 5 (9) 1554 (1988).

USING SIMULATED ANNEALING IN THE DESIGN OF COVERING CODES

PATRIC R. J. ÖSTERGÅRD
Digital Systems Laboratory
Helsinki University of Technology
02150 Espoo, FINLAND

Abstract

Simulated annealing has turned out to be a very efficient method in the search for covering codes. We discuss three methods, all using simulated annealing, for finding such codes. The direct approach and the approach using matrices have earlier been applied to some extent. The third method, the use of simulated annealing in finding acceptable partitions for seminormal codes, is novel. All these methods have led to new codes, better than any other known in the literature.

I. INTRODUCTION

Covering codes originated back in 1948, when Taussky and Todd [10] discussed the problem group-theoretically. During the years many contributions to the research have appeared in different journals within the areas of coding theory and combinatorics. The main interest has concerned binary codes and ternary codes of covering radius 1 (the so called *football pool problem*, see e.g. [8, 11]). There has been considerable recent growth in the interest in the area (cf. [3]).

The problem of finding covering codes can be formulated as a combinatorial optimization problem. *Simulated annealing* (SA) has turned out to perform amazingly well in the search for such codes.

II. COVERING CODES

We consider the space F_q^n consisting of all words of length n and coordinates belonging to the set $\{0,\ldots,q-1\}$. A code $C \subseteq F_q^n$ covers F_q^n with radius R if every word in F_q^n is within *Hamming distance* R from some codeword in C. The Hamming distance between two words is the number of coordinates in which they differ. A q-ary code of length n that has M codewords and covering radius R is called a $(q,n,M)R$ code. We denote

$$K_q(n,R) = \min\{M \mid \text{there is a } (q,n,M)R \text{ code}\}.$$

Example 1. $F_2^3 = \{000,001,010,011,100,101,110,111\}$. The code $C = \{000,111\}$ is a $(2,3,2)1$ code, since the words within Hamming distance 1 from 000 are $C_1 = \{000,001,010,100\}$ and from 111 are $C_2 = \{111,110,101,011\}$, and $C_1 \cup C_2 = F_2^3$. $K_2(3,1) = 2$, since there is no $(2,3,1)1$ code as is easily seen.

Codes with different kinds of partitioning properties have turned out to be very useful in constructing new codes from old ones. The concepts of *normal* and *seminormal* codes were introduced in [3] and [4], respectively, and in [13] the notion of *seminormal* codes was presented. In this paper we show have simulated annealing can be adopted to prove the seminormality of codes. We now define k-seminormal and strongly k-seminormal codes.

Definition 1: (Östergård, [13, Definition 2]) A $(q,n,M)R$ code C is said to be *k-seminormal*, if there is a partition of C into k subsets $C_0,\ldots C_{k-1}$ such that, for all $x \in F_q^n$, with $d(x,C) = R$,

$$\max_a\{d(x,C_a)\} \leq R+1.$$

Such partitions are called *acceptable.* The definition of strongly k-seminormal codes is obtained by removing "with $d(x,C) = R$" from Definition 1. The following theorem shows how seminormal codes can be used in the construction of new covering codes.

Theorem 1: (Östergård, [13, Theorem 3]) If there is a $(q,n,M)R$ seminormal code and q is a prime power, then there is a $(q,n+q,q^{q-2}M)R+1$ code.

Strongly seminormal codes are also of importance in constructing covering codes. These constructions are not presented here, but the interested reader is referred to [13].

III. SIMULATED ANNEALING

The algorithm used in our research is only a slight modification of the original simulated annealing algorithm presented by Kirkpatrick *et al.* [6, 7].

In the presentation of the methods that have been used we focus our attention on the definition of the neighbouring configurations and the energy function. Appropriate cooling schedules can usually be determined by some practical experiments.

IV. THE METHODS

Although we try to minimize the number of codewords covering a certain space (to determine $K_q(n,R)$), the methods to be presented here attempt to find coverings with a certain number of codewords. The minimization is attained by sequent attempts (runs) to find coverings with fewer and fewer codewords (the direct approach).

A. The Direct Approach

The first results concerning the use of simulated annealing in coding theory were presented by El Gamal *et al.* in 1987 [2]. In that paper covering codes were not considered, however, later the same year these were treated by Wille in [11].

The implementation of the direct approach is quite straightforward. The energy function is simply the number of words not covered by the codewords. When we are searching for a code C that covers F_q^n with radius R the energy function is

$$E = |\{x \in F_q^n \mid d(x,C) > R\}|.$$

In the annealing process we go through all codewords. A neighbourhood configuration is obtained by replacing a codeword c with a randomly chosen word c', such that $1 \leq d(c,c') \leq R$. The codewords could also be replaced with completely random words, however, experiments have showed that this way of defining the neighbourhood configuration is inferior to the previously mentioned one.

It has turned out that for covering radii $R = 1,2,3$ codes with up to about 100, 40 and 15 codewords, respectively, can be found within reasonable time. Even an ordinary personnel computer can be used for this (most of our work has been done on a 10 MHz PC/AT-compatible computer). With an increasing number of codewords the processing time required grows very fast, and the search for the $(3,7,186)1$ code found in [8] required many attempts and an extremely slow cooling rate. In the binary case the following theorem can be used to overcome this problem in some cases.

Theorem 2: (Östergård, [12, Theorem 2]) Let $C \subseteq F_4^q F_2^b$ be a code of covering radius R that has n codewords. Then there is a $(2,b+3q,2^q n)R$ code.

In the following subsections two other methods for finding coverings with many codewords are discussed.

B. Covering Using Matrices

The method to be presented in this section was first considered by Kamps and van Lint [5] and later generalized by Blokhuis and Lam [1]. In [9] van Lint Jr. generalized it to arbitrary covering radii.

Let $A = (I;M) = (a_1,\dots,a_n)$ be a $r \times n$ matrix where I is the $r \times r$ identity matrix and M is a $r \times (n-r)$ matrix with entries from F_q. For $s \in F_q^r$ the *R-covering of s using* A, $S_{A,R}(s)$, is defined as

$$S_{A,R}(s) = \{s + \sum_{j=1}^{n} \alpha_j a_j \mid \alpha_j \in F_q, |\{\alpha_j \neq 0\}| \leq R, 1 \leq j \leq n\}.$$

Consequently, A = I corresponds to covering in the traditional sense. A subset S of F_q^r *R-covers* F_q^r *using* A if

$$F_q^r = \bigcup_{s \in S} S_{A,R}(s).$$

Theorem 3: (van Lint, Jr., [9, Theorem 1.4.4]) If S R-covers F_q^r using a $r \times n$ matrix $A = (I;M)$, then $W = \{w \in F_q^n \mid Aw \in S\}$ covers F_q^n with radius R. $|W| = |S| \cdot q^{n-r}$.

The optimization problem now consists of finding the words in S and the M part of the matrix A. The interconnection between the codewords and the matrix causes problems, and it is not immediately clear how to perform the search efficiently. We consider three different ways to deal with this problem.

1. If there is a small number of nonequivalent matrices M it is possible to do the annealing in trying to find a set S for all these possibilities. This is the approach in [14]. A considerable speed-up can be achieved by performing the first run with a fast cooling rate, after which values of M leading to bad coverings can be removed. Repeating this procedure with slower and slower cooling rates and less and less numbers of values of M possibly leads to a covering. This approach has turned out to perform very well, however, it can not be applied when there is a big number of nonequivalent matrices M.
2. In the annealing process all words in S are gone through in one round (like in the direct approach), and after that attempts are made to change the columns of M with other columns. This is the approach in [8]. The performance of this approach has not turned out to be be very satisfactory.
3. We now describe how simulated annealing can be used *on two levels* to attack the problem. We first try to find a good value of M. For a certain M we try to find a good covering by searching for a set S. This is done using simulated annealing. Neighbourhood configurations are now obtained by replacing the columns of M with columns that differ in one position. These changes are accepted in the normal way depending on the number of uncovered words remaining after the next cooling. Having fixed M attempts are made to determine the minimal size of S.

C. Partitioning Seminormal Codes

We consider a $(q,n,M)R$ code C and describe a procedure that can be used in trying to prove k-seminormality of the code. An acceptable partition is found at the same time.

We divide the M codewords (preferably randomly) into k sets. If k divides M, the size of the sets is M/k, otherwise the size of some sets is $\lfloor M/k \rfloor$ and of some sets $\lceil M/k \rceil$. These sets are named $C_0,\dots,C_{k-1}$.

The energy function can now be taken as

$$E = |\{(x,a) \in (F_q^n, \{0,\dots,k-1\}) \mid d(x,C_a) > R+1, d(x,C) = R\}|.$$

In the annealing process we go through all the codewords of all the sets and the random displacement consists of changing the codeword in question and a random codeword *in another set*.

If we want to prove strong k-seminormality, only a slight change to the energy function has to be made.

V. CONCLUSIONS

Simulated annealing has turned out to perform surprisingly well in the search for covering codes, as a matter of fact probably no other area of coding theory has profited so much by this method. This can partly be explained by the fact that for example packing (error-correcting) codes are very structural of their nature and so for instance algebraic methods can be applied in the constructions. However, many known record-breaking coverings (the (3,6,73)1 code found in [8] to mention one) seem to possess no pattern at all.

In spite of the good results reported here, there is still some work to be done on this subject. This concerns especially the method of covering using matrices, with its special problems.

ACKNOWLEDGMENT

This work has been financially supported by the Finnish Academy.

REFERENCES

[1] A. Blokhuis and C. W. H. Lam, "More coverings by rook domains," *J. Combin. Theory*, vol. 36A, pp. 240-244, 1984.

[2] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 116-123, Jan. 1987.

[3] R. L. Graham and N. J. A. Sloane, "On the covering radius of codes," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 385 401, May 1985.

[4] I. S. Honkala, "Lower bounds for binary covering codes," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 326-329, Mar. 1988.

[5] H. J. L. Kamps and J.H. van Lint, "A covering problem," in *Combinatorial Theory and its Applications*, P. Erdös, A. Rényi, and V. T. Sós, Eds. Amsterdam: North-Holland, 1970, pp. 679-685.

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *IBM Research Report RC 9355*, 1982.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, May 1983.

[8] P. J. M. van Laarhoven, E. H. L. Aarts, J. H. van Lint, and L. T. Wille, "New upper bounds for the football pool problem for 6, 7 and 8 matches," *J. Combin. Theory*, vol. 52A, pp. 304 312, 1989.

[9] J. H. van Lint Jr., "Covering radius problems," M.Sc. thesis, Eindhoven University of Technology, June 1988.

[10] O. Taussky and J. Todd, "Covering theorems for groups," *Ann. Soc. Polon. Math.*, vol. 21, pp. 303-305, 1948.

[11] L. T. Wille, "The football pool problem for 6 matches. A new upper bound obtained by simulated annealing," *J. Combin. Theory*, vol. 45A, pp. 171-177, 1987.

[12] P. R. J. Östergård, "A new binary code of length 10 and covering radius 1," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 179-180, Jan. 1991.

[13] P. R. J. Östergård, "Upper bounds for q-ary covering codes", *IEEE Trans. Inform. Theory*, to appear in May 1991.

[14] P. R. J. Östergård, "New upper bounds for the football pool and other covering problems", in preparation.

A GENERAL PURPOSE DISTRIBUTED IMPLEMENTATION OF SIMULATED ANNEALING

R. Diekmann, R. Lüling, J. Simon
Department of Mathematics and Computer Science
University of Paderborn, Germany

Abstract: We present a problem independent general purpose parallel implementation of simulated annealing on distributed message-passing multiprocessor systems. We give a classification of combinatorial optimization problems. For typical representatives of the different classes good parallel simulated annealing implementations are presented. A new parallel SA-implementation is introduced. It works simultaneously on several markov chains and decreases the number of chains dynamically.

I. Introduction

Simulated Annealing (SA) was first presented by Kirkpatrick et al. [5] for solving hard combinatorial optimization problems and has proven to be a good technique to a lot of applications [3,4,6]. The disadvantage of this probabilistic approach is its large amount of computation time needed for obtaining a near-optimal solution. Some attempts at speeding up simulated annealing by using parallel computing systems have been made.

Two parallelization strategies are possible. In the first, problem describing data is distributed among several processors. This kind of parallelization depends on the given problem and is not easy to find for every optimization problem.

Therefore we consider a second strategy which is based on the parallelization of the actual simulated annealing algorithm. Every processor gets the complete problem instance and executes the sequential steps of the annealing algorithm in parallel. This approach is problem independent and easy to adapt. Some authors describe this technique using parallel systems with shared memory [1,3]. For special problems [2] distributed systems are used in the same way.

Aarts et al. [1] first decribed a technique where all processors work in parallel on the evaluation of one markov chain. With this idea we obtain a speedup of 23 with 64 processors when using a selfadapting cooling schedule.

We present an improved parallel SA algorithm that provides a speedup of 41 on 121 processors using a selfadapting cooling schedule. In this approach all processors start working on a number of distinct markov chains. The number of chains is reduced dynamically until all processors evaluate only one chain.

All algorithms are implemented in OCCAM-2 on a full reconfigurable transputer system.

II. Classification

The optimization problems are classified according to their evaluation time for the different steps of the sequential annealing algorithm. The most important two classes are:

1. problems where the generation of new configurations and the cost evaluation take equal time (e.g. TSP).
2. problems where the cost evaluation is much more expensive than the generation of configurations (e.g. placement).

III. Parallelization

We examined several parallelization approaches known from literature concerning their suitability for problems of the different classes. Kirkpatrick's ideas of locking parts of the configurations [3] were found to be inefficient when implemented in large distributed systems because no fast locking-mechanism is available.

Baiardi's model of a processor farm [2] provides nearly optimal speedup when used for problems of the second class. In this approach new configurations are generated by one master process and sent to a number of slave processes for cost evaluation and acceptance decision. If this method is applied to problems of the first class, the master is a bottleneck. He is not able to generate enough new configurations to keep a large number of slaves working.

III.a One-Chain

To get rid of this bottleneck new configurations must be generated by the slaves themselves. Only system updates are still controlled by the master. A version of this idea was described by Aarts et al. and implemented on a small parallel machine with shared memory [1].

In our implementation all slaves work on the evaluation of one markov chain If a slave detects an acceptable configuration the master is informed. He initiates a system update.

For convergence properties it is not profitable to use the first detected acceptable configuration for a global update. In a synchronization phase the master collects all accepted configurations and chooses one for a global update. Quickly calculated transitions are not favored.

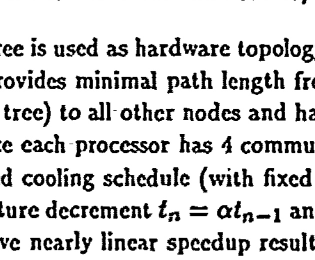


Fig.1: Results of One-Chain with a selfadapting cooling schedule

A ternary tree is used as hardware topology for our implementation. It provides minimal path length from the master-node (root of the tree) to all other nodes and has a maximal degree of four (since each processor has 4 communication links).

Using a fixed cooling schedule (with fixed initial temperature t_0, temperature decrement $t_n = \alpha t_{n-1}$ and fixed chain length L) we achieve nearly linear speedup results.

A selfadapting cooling schedule according to Huang et al. [4] is implemented. For the evaluation of the standard deviation ρ we use a smoothing technique similar to Otten [6].

Measurements of speedup and efficiency are shown in fig. 1. With this technique, it is not profitable to use more than 40 processors. The convergence behavior is equal to that of the sequential algorithm.

III.b Par-Chain

The behavior of the One-Chain is worse in the presence of high temperatures when implemented in large distributed systems. Many synchronization phases appear because almost every new configuration is accepted. For that reason no linear speedup is achieved.

To overcome this disadvantage several markov chains can be used and speedup is achieved by shortening the length of the individual chains. To guarantee convergence the chain length cannot be reduced arbitrarily. Since at lower temperatures the behavior of the One-Chain is much better, the idea was to combine these two methods [1].

When the algorithm starts all processors calculate their own markov chain according to the One Chain method (fig. 2). After a certain time a global synchronization step is made. All processors send their configurations to the root-node which controls the masters of One Chain. It chooses one of the configurations for further work.

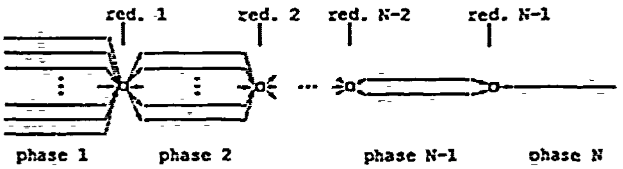


Fig.2: Reduction of number of chains

In case of low acception-rates, processors are clustered dynamically. This is done by building subtrees (fig. 3). Each cluster now calculates one chain according to the One-Chain implementation. While the number of chains is decreased, the chain length is magnified. The combination steps are repeated until all processors compute only one markov chain (fig. 2).

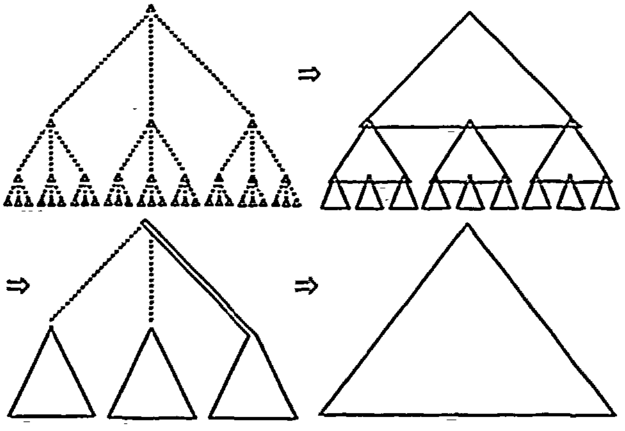


Fig.3: The phases of reduction (example with 40 processors).

The cooling schedule is controlled by the root-node. Calculation of initial temperature and temperature reduction are done in the same way as in the One-Chain implementation. The equilibrium detection cannot be done depending on the values of accepted configurations (like in One-Chain) because not all values from all clusters are available on the root-node. Therefore L is calculated before the computation of a chain starts according to a technique used by Otten [6].

Measurements of speedup and efficiency can be seen in fig. 4.

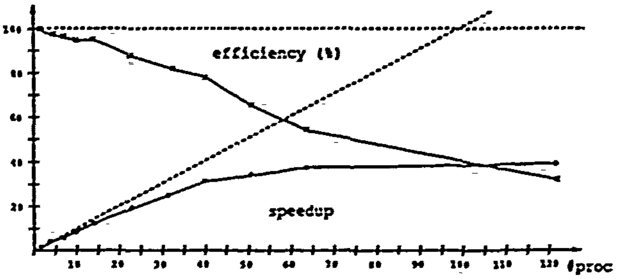


Fig.4. Results of the Par-Chain implementation

IV. Conclusion and further work

We presented a general purpose parallel implementation of SA on distributed multiprocessor systems. This approach is problem independent and easy to adapt. Hence it can be used as a tool of universal application for approximation of hard combinatorial optimization problems. To our knowledge it is the first problem independent parallelization of SA for large distributed systems.

Our next step is to combine methods from genetic algorithms [7] with the given SA parallelizations. Better convergence behavior is achieved by using a pool of markov chains. In the global synchronization phase of the Par-Chain algorithm the root-node chooses a pool of configurations for further work. The members of the pool and their rate of reproduction are selected concerning their 'goodness'. First experiments indicate also a better behavior in speedup.

References

[1] E. Aarts, F. de Bont, E.Haberts, P. van Laarhoven: *Parallel Implementations of the Statistical Cooling Algorithm.* North-Holland INTEGRATION, the VLSI journal 4(1986), pp. 209-238

[2] F. Baiardi, S. Orlando: *Startegies for a Massively Parallel Implementation of Simulated Annealing.* Parallel architectures and languages, PARLE '89, pp. 273-287

[3] F. Darema, S. Kirkpatrick, V.A. Norton: *Parallel algorithms for chip placement by simulated annealing.* IBM Journal of Research and Development, Volume 31, May 1987, pp. 391-402

[4] M.D. Huang, F. Romero, A. Sangiovanni-Vincentelli: *An Efficient General Cooling Schedule for Simulated Annealing.* IEEE International Conference on Computer Aided Design 1986, pp. 381-384

[5] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi: *Optimization by Simulated Annealing.* Science, Volume 220, May 1983, Number 4598, pp. 671-680

[6] R.H.J.M. Otten, L.P.P.P. van Ginneken: *The Annealing Algorithm.* Kluwer Academic Publishers 1988

[7] H. Muehlenbein, M. Gorges-Schleuter, O. Kraemer: *Evolution algorithms in combinatorial optimization.* Parallel Computing 7 1988, pp. 65-85

Parallelizing SA For Graph Embedding Is Hard *

John E. Savage
Department of Computer Science, Box 1910
Brown University
Providence, RI 02912
jes@cs.brown.edu

Markus G. Wloka
Department of Computer Science, Box 1910
Brown University
Providence, RI 02912
mgw@cs.brown.edu

Abstract We show that local search heuristics for grid and hypercube embeddings based on the successive swapping of pairs of vertices, such as simulated annealing, are P-hard and unlikely to run in polylogarithmic time. We have developed and implemented on the Connection Machine CM-2 a new massively parallel heuristic for such embeddings, called the *Mob* heuristic, which gives excellent results in practice.

1 Introduction

Graph embedding is the NP-complete problem of mapping a graph into another graph, called a *network* for clarity, while minimizing a cost function on the embedded edges of the graph. *Graph embedding* has application in VLSI (Very Large Scale Integration) placement and the minimization of data movement in parallel computers. An automatic graph-embedding tool optimizes communication resources, permits fault tolerance, and allows parallel programs to be divorced to some extent from the structure of the underlying communication network.

In this paper, we address *grid* and *hypercube embeddings*. The cost of a grid embedding is the sum of the *half-perimeters* of the boxes enclosing each edge. The cost of a hypercube embedding is the sum of the *Hamming distances* between the two vertices of each edge. Neither cost function measures routing congestion, but reduction of edge lengths can reduce congestion as a secondary effect.

Local search heuristics, of which steepest descent, Kernighan-Lin [2], and simulated annealing [3] are well-known examples, have been established as the heuristics of choice for general graph-embedding problems. The recent availability of general-purpose parallel processing hardware and the need to solve very large problem instances have led to increasing interest in parallelizing local search heuristics.

In local search algorithms, an initial solution is constructed, usually by some random procedure, and the cost function f is computed. Changes are made to the current solution, and the new solution, which we say is in the *neighborhood* of the current solution, replaces the current solution. The *SWAP neighbornood* of an embedding is the set of embeddings obtained by swapping the embedding of two vertices.

2 P-hardness Results

We show that local search algorithms for grid and hypercube embeddings are expected to be hard to parallelize by reducing them to the *circuit value problem* (CVP), the canonical P-complete problem.

Problems in NC can be solved on a parallel machine with polynomially many processors in polylogarithmic time. If a P-complete problem is in NC, then P, the class of problems solvable in polynomial time, is contained in NC, a highly unlikely result. Since logspace-reducibility is transitive, if a problem A is P-complete and we can find a logspace reduction of it to another problem B in P, then B is also P-complete. The definition of P-hardness does not require that the decision problem be in P. A P-hard problem is at least as hard to solve as a P-complete problem.

CVP is the problem of computing the value of a Boolean circuit from a description of the circuit and values for its inputs. CVP is P-complete [4]. Monotone CVP, a restricted version of CVP which uses only the operations AND and OR, is also P-complete [1].

Graph partitioning is a special case of graph embedding. The *graph-partitioning problem* is to partition the vertices of an undirected graph $G = (V, E)$ ($|V|$ even) into two sets of equal size such that the number of edges between them is minimized. We use the following result to give reductions to local search algorithms for grid and hypercube embeddings.

Theorem 1 *For the graph-partitioning problem, local search under the Kernighan-Lin neighborhood is P-complete. Local search under the SWAP neighborhood is P-complete, or P-hard if the local search algorithm is randomized [5,7].*

The result that local search under the SWAP neighborhood for graph partitioning is P-complete was independently obtained by Yannakakis and Schäffer [9].

*This work was supported in part by the National Science Foundation under Grant CDA 87-22809 and the Office of Naval Research under contract N00014-83-K-0146 and ARPA Order Nos. 4786, 6320.

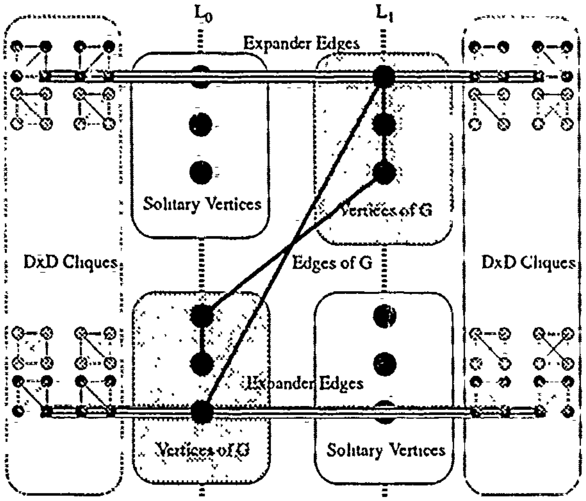


Figure 1: Graph partition to grid embedding reduction.

The graph-partitioning completeness proof can be summarized as follows. A graph G is constructed from a monotone boolean circuit C. The graph G contains subgraphs corresponding to the AND and OR gates in the circuit C as well as auxiliary subgraphs. A partition of G into equal-sized sets is given as a starting point. By applying improving swaps until a local minimum is reached, the value of the circuit C can be computed directly from the resulting graph partition.

Theorem 2 *For the graph-embedding problem on the grid and hypercube, local search under the SWAP neighborhood is P-complete, or P-hard if the local search algorithm is randomized.*

To prove Theorem 2, we give two logspace reductions that map the graph G used in the graph-partitioning proof into initial embeddings in the grid and hypercube. The full description of the constructions are given in [6]. We give a sketch here of the grid embedding construction, the construction for the hypercube embedding problem is similar.

For the grid embedding problem, the construction shown in Figure 1 consists of a grid with two vertical lines L_0, L_1 which have unit horizontal separation. The line L_0 represents a logical value of 0 while the other line represents a logical value of 1. Let (X_0, Y_0) be the initial partition of G. Let K be the maximum degree of G. $(K = 9)$[5]. The vertices in the set X_0 of G are placed on L_0 as shown in Figure 1 and above them is an equal number of solitary vertices. The vertices in Y_0 are placed on L_1 above an equal number of solitary vertices. These vertices on the vertical lines L_0 and L_1 are spaced D vertical grid points apart, where $D = K + 1$, to leave enough room for the rest of the construction. With this construction we show that the only swaps accepted by the local search heuristic are between a vertex of G and the opposing solitary vertex.

To limit the movement of a vertex v in G, we place L $(L = 2K + 2)$ $D \times D$ *anchor cliques* to the left and another L anchor cliques to the right of the two vertical lines, and we connect the closest vertex of every clique to v with an *expander edge*, of which there are $2L$. The solitary vertices are not anchored, and can thus move freely. We shall show that the expander edges and cliques constrain v to move only along the x-axis between the two vertical lines.

Both constructions have the property that all possible positive gain swaps in this problem correspond to positive gain swaps in the associated graph-partitioning problem, and thus mimic the computation of the value of a circuit. We find that any local-search based algorithm is P-hard if every swap accepted by it corresponds to a positive gain swap in the graph-partitioning algorithm.

Since local search heuristics based on the SWAP neighborhood for grid and hypercube embeddings are P-hard, it is unlikely that a parallel algorithm exists that can find even a local minimum solution in polylogarithmic time in the worst case. This result puts experimental results reported in the literature into perspective: attempts to construct the exact parallel equivalent of serial simulated-annealing-based heuristics for graph embedding have yielded disappointing parallel speedups.

3 The *Mob* Heuristic

We have developed a new massively parallel heuristic, which we call the *Mob* heuristic. The heuristic is closely related to both Kernighan-Lin and simulated annealing. The algorithm uses a *mob-selection rule* to swap large sets of vertices, called *mobs*, across planes of a grid or hypercube. If the new embedding found has a smaller cost, the search is repeated on the new embedding with the same mob size. If the cost increases, the neighborhood of the new embedding is searched with a smaller mob size. We assume that *Mob* executes a number of *iterations* that does not exceed a polynomial in the number of graph vertices. A "schedule" determines the rate at which the mob size decreases. The P-hardness result given above applies also to the *Mob* heuristic if the mob size is fixed at one.

The mob-selection rule searches for an approximation to the subset that causes the largest improvement in the embedding cost and is designed to be computed very quickly in parallel. On the hypercube a hyperplane is chosen at random and vertices are swapped between hypercube neighbors across the chosen hypercube axis. On the grid, a distance d of $\pm 1, 2, 4, 8, 16, \ldots$ on either the X or Y axis is chosen at random, and vertices are swapped between grid neighbors that are a distance d apart. In both cases a *gain* is computed for every vertex to find the vertex pairs whose exchange would cause the largest individual change in the embedding cost. A group of high-gain vertices of size *mob* is selected to move. To avoid sorting by gain, an expensive operation on a massively parallel machine, we use adaptive binary search to identify a set of vertices with gain g or larger in each set where g is the smallest gain such that at least *mob* vertices have a gain greater than or equal to g. We select *mob* vertices at random from this set.

We evaluated the performance of the *Mob* hypercube

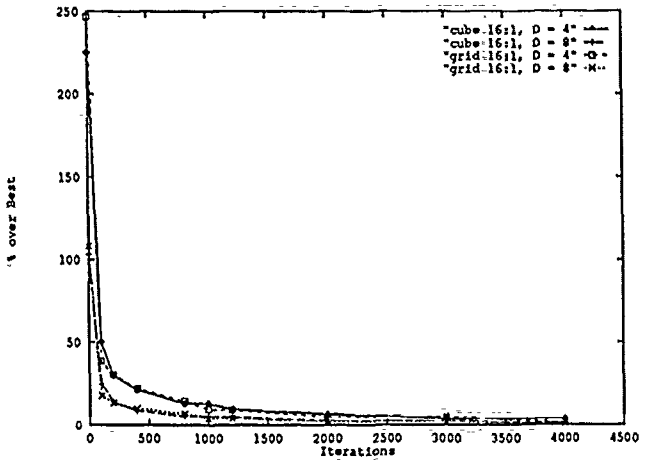


Figure 2: Convergence of *Mob*

and grid embedding algorithms by conducting an extensive series of experiments on the Connection Machine CM-2.

The CM-2 is a massively parallel SIMD (Single-Instruction-Multiple-Data) machine: each processor executes the same instruction at the same time unless it has been disabled. The SIMD approach simplifies debugging, permits an elegant programming style, and does not limit the expressiveness of algorithms. The up to 64K moderately slow one-bit processors of the CM-2 are organized as a 12-dimensional hypercube with sixteen nodes at each corner of the hypercube. The CM-2 supports virtual processors which are simulated very efficiently in microcode. Each processor has an associated memory of up to 8K bits. The memory is shared by passing messages among processors. The CM-2 also permits communication along hypercube and multidimensional grid axes, which is substantially faster than the general router.

Our experimental data is summarized here but reported elsewhere [8]. In our experiments, 1-to-1 mappings of graph vertices to network nodes were used to model VLSI placement problems and standard embedding problems. We also wanted to model bulk-parallelism, as described by Valiant [10], where n virtual processors are embedded into a computing network of size $n/\log n$, and chose 16-to-1 mappings as a rough approximation of $\log n$-to-1 mappings.

We conducted experiments on randomly generated graphs of small ($d = 3 \ldots 16$) degrees with up to 500K vertices and 1M edges! The number of iterations needed by the *Mob* heuristic to reach a fixed percentage above the best-ever embedding cost appears to be approximately constant, on the order of 4000 to 8000, and therefore independent of graph or network size. This holds for hypercube and grid embeddings, and for both 1-to-1 and 16-to-1 mappings. The rate of convergence is shown in Figure 2 for 16-to-1 mappings of the *Mob* grid and hypercube-embedding algorithms. A good solution is produced rapidly; further improvements can be obtained if enough computation time is available. The number of iterations to achieve a given percentage decreases as the degree increases. The *empirical* parallel complexity of the *Mob* heuristic on the CM-2 (on random graphs) is time $O(\log |E|)$ with $2|E|$ processors.

The absolute speed at which an embedding is produced shows that *Mob* can be implemented very efficiently on a SIMD-style machine. It takes approximately 36 minutes to find an embedding of a 500K-vertex, 1M-edge graph, the largest graph that would fit into the CM-2, into a 1024 x 512 grid. Due to excessive run times, previous heuristics reported in the literature were able to construct graph embeddings only for graphs that were 100 to 1000 times smaller than those used in our experiments. On small graphs, where simulated annealing and other heuristics have been extensively tested, our heuristic was able to find solutions whose quality was at least as good as simulated annealing.

References

[1] L. M. Goldschlager, "The Monotone and Planar Circuit Value Problems," *ACM Sigact News*, vol. 9, no. 2, pp. 25–29, 1977.

[2] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *AT&T Bell Labs. Tech. J.*, vol. 49, pp. 291–307, Feb. 1970.

[3] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[4] R. E. Ladner, "The Circuit Value Problem is Log Space Complete for P," *ACM SIGACT News*, vol. 7, no. 1, pp. 18–20, 1975.

[5] J. E. Savage and M. G. Wloka, "Parallelism in Graph-Partitioning," in *Journal of Parallel and Distributed Computing*, to appear, 1991.

[6] J. E. Savage and M. G. Wloka, "Parallel Graph-Embedding and the Mob Heuristic," Department of Computer Science, Brown University, Technical Report No. CS-91-07, 1991.

[7] J. E. Savage and M. G. Wloka, "On Parallelizing Graph-Partitioning Heuristics," in *Proceedings of the ICALP'90*, pp. 476–489, July 1990.

[8] J. E. Savage and M. G. Wloka, "Parallel Graph-Embedding Heuristics," in *5th SIAM Conference on Parallel Processing for Scientific Computing*, Houston, to appear, Mar. 1991.

[9] A. A. Schäffer and M. Yannakakis, "Simple Local Search Problems That Are Hard to Solve," *SIAM Journal on Computing*, vol. 20, no. 1, pp. 56–87, Feb. 1991.

[10] L. G. Valiant, "A Bridging Model for Parallel Computation," *Communications of the ACM*, vol. 33, no. 8, pp. 103–111, Aug. 1990.

The Effect of Structure in the Mapping Problem Using Simulated Annealing*

Craig Lee
Lubomir Bic
Department of Information & Computer Science
University of California
Irvine, CA 92717, USA

Abstract

In parallel computation, a common requirement is the mapping of a problem graph of communicating tasks onto a network graph of processing elements such that (1) the communication distance is minimized, and (2) the problem graph is evenly distributed over the network. When using simulated annealing on this problem, how does the relative structure of the two graphs affect performance? To investigate this question, we used simulated annealing to map four graph types of varying "structure", representing possible problem graphs, onto a graph with a very regular structure, representing a network of processing elements. These results show that higher regularity produces (1) a higher but narrower range of final energy values, (2) a higher distance-energy correlation and (3) a higher degree of ultrametricity.

1 Introduction

Simulated Annealing is a computational technique of finding a near-optimal "solution" to an optimization problem by making random changes in the solution and probabilistically accepting the change depending on whether it improves or degrades the solution [3, 1]. How well simulated annealing performs is dependent on the *energy landscape* of the *configuration space*. How can we determine the character of the landscape for a given configuration space? Solla et al. hypothesize that configuration spaces exhibiting a high degree of *ultrametricity* are more suitable for annealing [7]. Can we use ultrametricity and the correlation between energy and distance to determine if annealing will perform well on a given class of optimization problems? Concomitantly, how does the "structure" of a class of problems affect the degree of ultrametricity and performance? We will investigate these questions by mapping four different graph types that represent problem graphs with a range of structure onto a regular network graph.

2 Four Graph Types

The energy metric we use here for annealing is the distance-variance function as defined in [1]. This attempts to spread a problem graph evenly over the network graph while minimizing communication distance by weighting the two metrics of communication distance and the variance of the number of problem graph edges that can be said to be incident on a network graph vertex as a result of mapping. Ultrametric correlation coefficients are computed as described in [7]. We now investigat: the effect of problem structure on the behavior of simulated annealing and quenching by mapping members of four different graph types, (1) random, (2) geometric, (3) torus, and (4) n-cube, onto a torus. These graph types were chosen to represent a range of "structure". The random graph can be said to have no structure. The geometric graph, as defined below, has more structure with similarities to a torus. The torus and n-cube are very regular in their structure but are similar and dissimilar, respectively, to the network graph.

These graphs are defined in the following way. All problem graphs have 32 vertices and an average vertex degree of 5. This ensures that any difference in numerical energy results from differences in graph structure, i.e, how well the graph can be mapped, rather than just the number of vertices and edges involved. A random graph is generated by defining n vertices and connecting any two with probability p which results in an average degree of $(n-1)p$ [2]. Given the desired degree of 5, it is easy to compute the required p. A geometric graph is generated by randomly distributing n vertices over the unit square and connecting any two vertices that are contained within a square with side $0 < k < 1$ where this smaller square can wrap-around to the other sides of the unit square. This definition is intended to produce a topology similar to a torus. (If wrap-around was not allowed, the generated graph would be more similar to the layout of a printed circuit board as defined in [2].) Here the average degree is $4nk^2$. Given the desired degree of 5, it is easy to compute the required k. The problem torus will be a 4×8 grid but with extra edges added in a symmetric manner such that each vertex has a degree of 5. Finally, the n-

[*] A full-length version of this paper is available from the first author at lee@aerospace.aero.org.

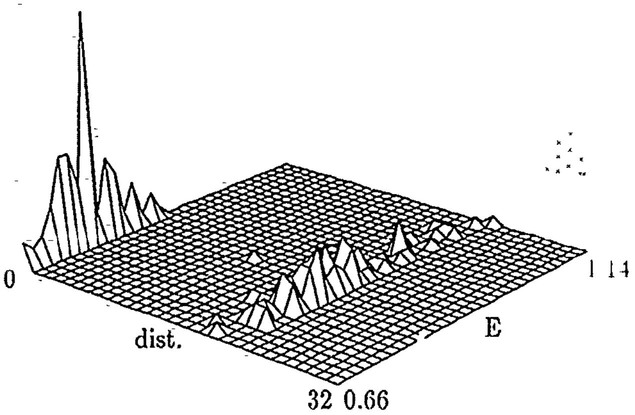


Figure 1: Distance-Energy Surface; Random

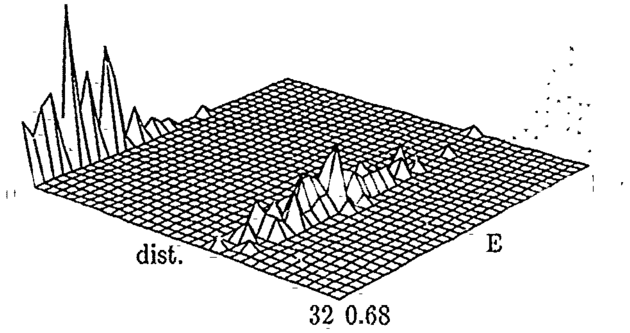


Figure 2: Distance-Energy Surface; Geometric

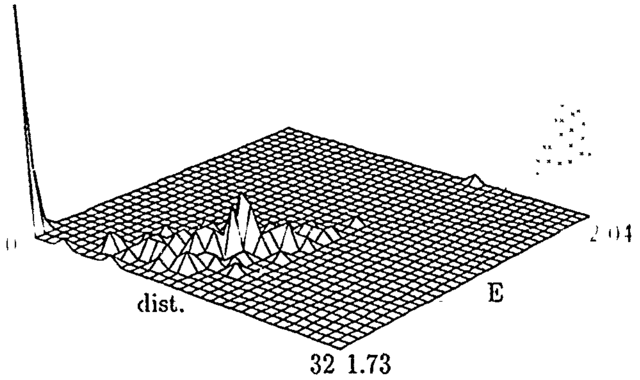


Figure 3: Distance-Energy Surface; Torus

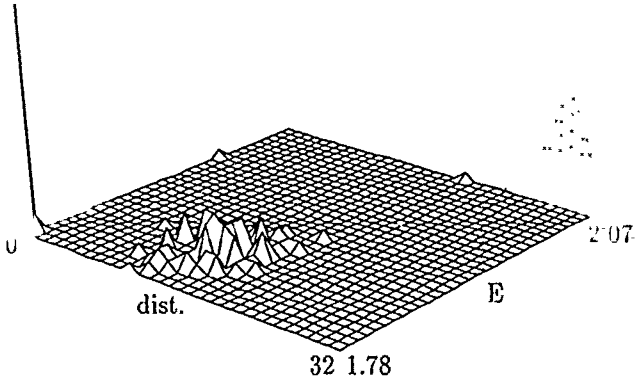


Figure 4: Distance-Energy Surface; n-Cube

cube with 32 vertices naturally has a degree of 5. Finally, the network graph will be a 3×3 torus, each vertex will have a degree of 4 in the typical grid fashion. This network graph may seem small but it is not a degenerate torus and it was deemed necessary to hold down the cost of checking permutations. All of these graphs may seem small but each configuration space has a size of 9^{32}.

For each problem graph, we will do 100 slower annealings and 100 faster annealings (called quenchings) onto the network graph. In all cases, the same initial mapping will be used but with a different seed for the random number generator. In all cases, an even 0.5/0.5 weighting will be used in the distance-variance energy function. The *temperature* control parameter has an initial value of 100 ($\approx 50\times$ the initial energy value) and an exponential decay factor of 0.9. At each temperature, thermal equilibrium will be determined as in [3]. for a problem of size n, equilibrium is reached after an changes have been accepted or bn changes have been attempted (accepted or not), whichever comes first. The user is free to choose a and b under the constraint that $0 < a \leq b$. For quenching, $a/b = 1/2$. For annealing, $a/b = 128/256$. The algorithm terminates when the configurations at the last three consecutive temperatures show no improvement, i.e., the configuration is "frozen" [5].

For each quenched solution within each graph type, the map permutation with the minimum distance to any annealing solution is found. We then plot the distance versus the energy as a histogram surface as shown in Figures 1 to 4. All annealed solutions are set at distance zero and the distance axis shows how far away the closest quenched solution permutations are. Since the problem graphs all have 32 vertices, all distances are in the range $\langle 0, 32\rangle$. The energy axis, however, is scaled to the minimum and maximum values of all annealings and quenchings and then quantized into 32 bins. The vertical axis then shows the number of solutions in a particular bin. The vertical scale is arbitrary but uniform across all graphs. This graphical display shows at a glance the distribution of the annealed and quenched solutions relative to one another. For comparison, the general distance distribution of all possible configurations from any one configuration is shown as a dotted curve.

Figures 1 to 4 show the following. The random and geometric graphs have very similar results, there is a significant range of annealed energies and an even broader range of energies for the quenched solutions. The additional "structure" in the geometric graph did not significantly change the distribution of solutions found. All of the quenched solutions are in a narrow range of distances from their closest annealed solutions. There is no apparent correlation between the distance between solutions and their difference in energy. In fact, the distribution of quenched solutions seems strongly influenced by the probability of finding a close configuration within the general distribution of configurations.

The torus and n-cube graphs, however, tell a different

Graph Type	Quenching	Annealing
random	0.028	0.036
geometric	0.033	0.035
torus	0.238	0.401
n-cube	0.167	0.222

Table 1: Ultrametric Correlation Coefficients

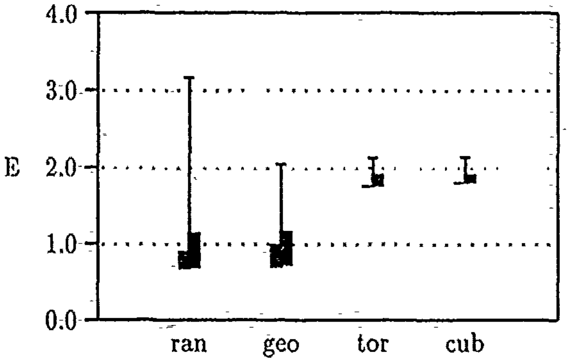


Figure 5: Range of Absolute Energy Values

story. The annealed solutions have a tightly-clustered energy range. (For the sake of typesetting, the peaks in these surfaces have been truncated.) The quenched solutions show a definite correlation between distance and energy aside from a few solutions trapped in high local minima. The ultrametric correlation coefficients were computed as discussed in the previous section and are shown in Table 1. This clearly supports the notion that simulated annealing performs better in a configuration space that exhibits ultrametricity which is, in turn, related to the correlation between distance and energy difference. What the graph types tell us is that this correlation is dependent on the regular and self-similar structure present in the problem.

The differences discussed so far, however, are *relative* differences. Figure 5 shows the absolute energy range for quenching and annealing for the four different graph types including the initial map energy. This shows that while the random and geometric graphs start at a equal or much higher energy, they finish in an energy range that is broader and much lower than the torus and n-cube. (Note that these bars exclude solutions trapped in high local minima.) What this tells us is that graph structure affects the size and distribution of local minima that trap annealing and quenching.

3 Summary and Conclusions

These experiments have shown that higher regularity in the problem graph produces (1) a narrower but higher range of final energy values, (2) a correlation between move distance and energy, and (3) a higher ultrametric correlation. The fact that quenched solutions with a higher energy can't find a closer annealed solution implies that the configuration space itself has a regular shape, that "valleys" with the same shape have multiple occurances such that a quenching that lands with energy E in one valley may be distant from an annealing with energy $E-\epsilon$ in another valley but has a closer isomorphic permutation in that same valley. Such an interpretation is consistent with the argument given by Solla et al. in support of the correlation between move distance and ultrametricity. However, it is an open question whether these configuration spaces have 'self-similar' or 'fractal-like' valleys, as described in [6], such that similarly-shaped valleys occur with a recursively decreasing size. The observations discussed so far lead us to conjecture that regularity in any optimization problem produces a configuration space that reflects the same structure in which low-lying minima are ultrametrically distibuted. This implies that when higher regularity exists, a shorter annealing schedule can be used with a higher probability that the resulting maps will not be far apart in energy.

References

[1] D.H. Ackley, "Stochastic Iterated Genetic Hillclimbing," PhD thesis, Carnegie Mellon University, CMU-CS-87-107, March, 1987.

[2] J.R. Anderson, C. Peterson, "Applicability of Mean Field Theory Neural Network Methods to the Graph Partitioning Problem," *MCC Tech. Rep. ACA-ST-064-88*, February, 1988.

[3] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, 220.671-680, May 13, 1983.

[4] C.A. Lee, "Logic, Parallelism and Semantic Networks: the Binary Predicate Execution Model," PhD thesis, Dept. of Information & Computer Science, University of California, Irvine, *Tech. Rep. #88-30*, December 1988.

[5] F. Romeo, A. Vincentelli, C. Sechen, "Research on Simulated Annealing at Berkeley," *IEEE Int'l. Conf. Computer Design*, pages 652-657, 1984.

[6] G.B. Sorkin, "Simulated Annealing on Fractals. Theoretical Analysis and Relevance for Combinatorial Optimization. In W.J. Dally, editor, *Advanced Research in VLSI*, pages 331-351. MIT Press, 1990.

[7] S.A. Solla and G.B. Sorkin and S.R. White, "Configuration Space Analysis for Optimization Problems," In Bienenstock et al., editors, *Disordered Systems and Biological Organization*, pages 283 293. Springer-Verlag, 1986.

PROPERTIES OF SIMULATED ANNEALING WITH INACCURATE COST FUNCTIONS

Daniel R. Greening*

ABSTRACT

Inaccurate cost or energy functions appear in many parallel simulated annealing implementations. Such errors alter the equilibrium cost, change the speed to approach equilibrium, and affect the quality of the final result.

We show that identically distributed Gaussian errors do not affect equilibrium. Non-id Gaussian errors and bounded errors can alter equilibrium; such changes vary with temperature and error magnitude. The convergence speed under bounded errors also depends on temperature and error magnitude.

Finally, we show that constraining errors to a constant factor of the temperature guarantees convergence in a restricted case.

1 INTRODUCTION

Simulated annealing is an algorithm to find nearly-optimal solutions to *NP*-hard problems [1]. Figure 1 outlines the approach: s_0 is the initial state, $G_{s,s'}$ is the generation probability, random returns an uniformly distributed random number from $[0,1]$, C is the cost function, and T is the temperature. The sequence of temperatures T_i is the temperature schedule. It is generally presumed that an optimum schedule keeps the average observed cost close to its equilibrium value (not necessarily true, see [2]).

```
$T \leftarrow T_0$;
$s \leftarrow s_0$;
while not done
    $s' \leftarrow$ generate($s$) with probability $G_{s,s'}$,
    $\Delta \leftarrow C(s) - C(s')$;
    if $(\Delta < 0) \vee (\mathtt{random}() < e^{-\Delta/T})$ then $s \leftarrow s'$;
    $T \leftarrow$reduce-temperature($T$);
    end while;
```

Figure 1. Simulated Annealing Algorithm

Many parallel implementations involve approximate cost functions [3]. Some sequential implementations also use approximate functions: examples include congestion and wire-length estimates in circuit placement [1].

1.1 Annealing Properties

Define acceptance matrix at time t, $A^{(t)}$, $A^{(t)}_{s,s'} = e^{\min[0,(C(s)-C(s'))/T_t]}$. Define inhomogeneous Markov chain P for simulated annealing,

$$P^{(t)}_{s,s'} = \begin{cases} G_{s,s'}A^{(t)}_{s,s'} & \text{if } s \neq s' \\ 1 - \sum_x \neq s' P_{t,s,x} & \text{otherwise} \end{cases}$$

If we fix the temperature, $A^{(t)} = A$, $P^{(t)} = P$, and the chain is homogeneous. If it is also ergodic, each state s has an equilibrium probability $\rho(s)$ independent of the initial state.

Existing programs attempt to bring state probabilities close to equilibrium at each temperature [1]. Thus, we can gain information about annealing behavior by looking at the homogeneous chain.

We can guarantee ergodicity if these properties hold.

$$\begin{array}{rll} \text{probability} & \forall s \in S, \sum_{s' \in S} G_{s,s'} = 1 & (1) \\ \text{coverage} & \forall s, s' \in S, \exists k \geq 1, [(G^k)_{s,s'} \neq 0] & (2) \\ \text{aperiodicity} & \exists s \in S, [P_{s,s} \neq 0] & (3) \\ \text{finiteness} & |S| \in \mathbf{Z}^+ & (4) \\ \text{symmetry} & \forall \langle s, s' \rangle \in S \times S, [G_{s,s'} = G_{s',s}] & (5) \end{array}$$

(1) states that G_s is a probability vector. (2) guarantees that the annealing chain is irreducible. (3) and (4) make the chain is aperiodic and finite. Irreducible, aperiodic, finite Markov chains are ergodic.

1.2 Inaccurate Cost Functions

Parallel algorithms can reduce communication costs by using some stale information, instead of maintaining elaborate synchronization to keep local information up-to-date. Stale data cause inaccuracies in the cost. Experiments show that allowing inaccuracies can improve speed, but can degrade results [4, 5].

Two questions arise: How does the result quality of asynchronous simulated annealing compare to that of sequential simulated annealing? How fast does asynchronous simulated annealing converge to a solution, compared to sequential simulated annealing?

Using a thermodynamic analogy, Grover showed the effect of bound ed errors on the partition function [6]. Durand and White analyzed equilibrium properties for bounded errors on a restricted algorithmic class [7]. Gelfand and Mitter showed that state-independent noise, under some conditions, will not affect asymptotic convergence [8].

We do *not* assume state-independence; most applications appear to exhibit strongly state-dependent errors. We expand the scope further by considering both fixed error bounds and Gaussian errors in the general case, and by examining the effect of bounded errors on speed.

Experiments have hinted that when errors are proportionally constrained to temperature, results improve. Invoking these observations, researchers have modified asynchronous algorithms to obtain better final states [9, 10]. Our last result validates their assumptions.

Proofs for equilibrium properties and convergence speed are omitted (see [11]). Proof of the final result, on convergence to global optima, is retained.

2 EQUILIBRIUM PROPERTIES

We refer to the true cost, $C(s)$, of state s, and the erroneous cost $C_\epsilon(s) = C(s) + \epsilon(s)$, of state s. ϵ is a random variable dependent on s, thus $C_\epsilon(s)$ is a random function.

2.1 Bounded Errors

If cost-function errors are bounded, then we have $\underline{\epsilon} \leq \epsilon \leq \overline{\epsilon}$.

*University of California, Los Angeles. dgreen@cs.ucla.edu.

Theorem 1 *Let $\rho_\epsilon(s,T)$ and $\rho(s,T)$ be the equilibrium probabilities of state s at temperature T, with bounded errors and without errors, respectively. Then,*

$$e^{(\underline{\epsilon}-\overline{\epsilon})/T}\rho(s,T) \le \rho_\epsilon(s,T) \le e^{(\overline{\epsilon}-\underline{\epsilon})/T}\rho(s,T) \tag{6}$$

Equilibrium behavior is often characterized by its "macroscopic properties." Any macroscopic property, $F(T)$, is the expected value of some function, $f(s)$, over the state space,

$$F(T) = \sum_{s\in S} f(s)\rho(s,T) \tag{7}$$

Theorem 2 *Let $F_\epsilon(T)$ and $F(T)$ be equivalent macroscopic properties, for function f, at temperature T, with cost functions C_ϵ and C, respectively. Then*

$$e^{(\underline{\epsilon}-\overline{\epsilon})/T}F(T) \le F_\epsilon(T) \le e^{(\overline{\epsilon}-\underline{\epsilon})/T}F(T) \tag{8}$$

A commonly measured macroscopic property, the average cost, is constrained by Theorem 2.

2.2 Gaussian Errors

Simulated annealing typically operates on structures with discrete cost functions: thus the errors appear as discrete values. However, as we add state variables and as the maximum number of uncorrected interdependent moves increases (through increased parallelism or slower update times), the errors can approach a Gaussian distribution.

In many instances, particularly in parallel applications, the probability distribution of the observed cost function $C_\epsilon(s)$ is reflected about the true cost $C(s)$. Cost-functions exhibiting this behavior appear in work by Durand and White [7].

Thus, it is reasonable to investigate the effect of Gaussian random cost function C_ϕ, with mean $E[C_\phi(s)] = C(s)$. We will show that when the variances of the state costs do not differ greatly, simulated annealing with inaccurate costs converges to a good solution.

Lemma 1 *Let the cost of each state s be $C_\phi(s) = C(s)+X_s$, where X_s is an independent random variable. Execute the simulated annealing algorithm in Figure 1 with lines 3 and 6 sampling random variables $C_\phi(s)$ and $C_\phi(s')$ instead of bounded random variables, and with T fixed. If (1)-(5) are satisfied and $T > 0$, the resulting homogeneous Markov chain P_ϕ is ergodic, and the equilibrium probabilities are given by*

$$\rho_\phi(i) = \frac{e^{-C(i)/T}E[e^{-X_i/T}]}{\sum_{j\in S} e^{-C(j)/T}E[e^{-X_j/T}]}. \tag{9}$$

Corollary 1 *If all X_s are identically distributed, and the resulting Markov chain is ergodic, then $\rho_\phi = \rho$ and $F_\phi = F$.*

Theorem 3 *Let $C: S \to \mathbf{R}$ be a cost-function, and let its stationary Boltzmann distribution be $\rho: S \to [0,1]$. Consider a random cost function $\phi: S \to \mathbf{R}$, where each random variable $\phi(s)$ is an independent Gaussian distribution with mean $C(s)$ and variance $\sigma^2(s)$. Let $\rho_\phi: S \to [0,1]$ give its stationary Boltzmann distribution. Then $\rho_\phi(s)$ can be bounded by*

$$e^{(\underline{\sigma}^2-\overline{\sigma}^2)/2T^2}\rho(s) \le \rho_\phi(s) \le e^{(\overline{\sigma}^2-\underline{\sigma}^2)/2T^2}\rho(s) \tag{10}$$

Corollary 2 *Macroscopic property F_ϕ is bounded by*

$$e^{(\underline{\sigma}^2-\overline{\sigma}^2)/2T^2}F(T) \le F_\phi(T) \le e^{(\overline{\sigma}^2-\underline{\sigma}^2)/2T^2}F(T) \tag{11}$$

3 EQUILIBRIUM MIXING SPEED

If P is a Markov chain and $\rho(i)$ is the stationary probability of state i, define the *conductance of a subset*, Φ_V, $V \in S$ as

$$\Phi_V = \frac{\sum_{i\in V, j\notin V}\rho(i)P_{ij}}{\sum_{i\in V}\rho(i)} \tag{12}$$

In words, the conductance of a subset V is the conditional probability that a transition will leave V, given that we start in V.

Let $\rho(V) = \sum_{v\in V}\rho(v)$, and let $S_{1/2} = \{V \subset S | \rho(V) \le 1/2\}$. Define the *global conductance* as the minimum conductance over all subsets with stationary probability below 1/2, thus,

$$\Phi = \min_{V\in S_{1/2}} \Phi_V \tag{13}$$

This global conductance provides a good measure for annealing speed It is related to another speed measure, the dominant eigenvalue [12].

Others have examined the effect of adjusting the move spaces to obtain better annealing speed, using conductance and eigenvalues [13, 14]. Here, we show how the size of the errors constrain the global conductance.

Theorem 4 *Consider two different annealing chains P and P_ϵ, with state space S and generation probability G identical. Annealing chain P has cost function C, and P_ϵ has cost function C_ϵ. They are related by $C(s)+\underline{\epsilon} \le C_\epsilon \le C(s)+\overline{\epsilon}$. Let Φ and Φ_ϵ be the corresponding global conductances. Then,*

$$e^{3(\underline{\epsilon}-\overline{\epsilon})/T}\Phi_\epsilon \le \Phi \le e^{3(\overline{\epsilon}-\underline{\epsilon})/T}\Phi_\epsilon \tag{14}$$

4 CONVERGENCE TO GLOBAL MINIMA

Simulated annealing can be made to converge monotonically to the optimum result, using an appropriate temperature schedule Define global optima set S_0 such that $s \in S_0 \Rightarrow C(s_0) = \min\{C(s')|s' \in S\}$

The most general results presume no particular structure to the state space, other than those specified by (1) (5). We pay a penalty in time for generality. Under these prepositions, a simulated annealing temperature schedule with $T(t) = c/\log t$ will converge to the minimum energy states [15, 16].

Proofs of convergence for these general spaces consider sets of local minima R^k, and presume the cost function returns an integer. Let R^0 be the set of all local minima. Roughly, $(R^k \setminus R^{k+1})$ is the set of local minima which can be escaped to a lower cost by ascending a change in cost of k (i.e., the height of the cup containing the local minimum). Thus, if the state set is finite and fully connected, there is some d such that $R^d \subset S_0$ (see [16] for a rigorous definition).

Theorem 5 *Suppose an annealing chain satisfies (1)-(5) and has cost function $C_\epsilon(s,t)$ with time and state dependent errors $\epsilon(t)$, such that*

$$C_\epsilon(s) + \underline{\epsilon}(t) \le C_\epsilon(s,t) \le C_\epsilon(s) + \overline{\epsilon}(t). \tag{15}$$

Let $b_1T(t) \le \underline{\epsilon}(t) \le \overline{\epsilon}(t) \le b_2T(t)$, where b_1 and b_2 are constants. Let the temperature schedule be of the form $T(t) = d/\log t$, where $d \in \mathbf{Z}^{0+}$. Then the algorithm converges in probability to the set of global minima if $R^d \subset S_0$.

Proof. Suppose the transition matrix P^t for some Markovian system at time t is constrained by (16).

$$c_1e^{-D^0_{ij}/T(t)} \le P^t_{ij} \le c_2e^{-D^0_{ij}/T(t)}, \tag{16}$$

where c_1 and c_2 are positive constants. Assume for some integer $d \ge 0$ that (17) and (18) hold.

$$\sum_{t=0}^{\infty} e^{-d/T(t)} = \infty \tag{17}$$

$$\sum_{t=0}^{\infty} e^{-d-1/T(t)} < \infty \tag{18}$$

We can then conclude that (19) and (20) are true [16].

$$\forall i \in S,\ \lim_{t\to\infty} P(X(t) \in R^d | X(0) = i) = 1 \tag{19}$$

$$\forall i \in R^d,\ \limsup_{t\to\infty} P(X(t) = i | X(0) = i) > 0 \tag{20}$$

Let $c_1 = e^{-b_1}$ and $c_2 = e^{-b_2}$. These values satisfy (15) and (16).

Choose d so that $R^d \subset S_0$. Such a d must exist, since (1)-(5) are satisfied. Let $T(t) = d/\log t$. This satisfies (17) and (18). By (19) the erroneous simulated annealing algorithm converges in probability to the set of global minima. ■

5 CONCLUSION

Calculation errors in simulated annealing affect the equilibrium cost and the speed at which equilibrium is reached. Analytic results presented in this manuscript constrain equilibrium properties with a bounded or Gaussian random cost function. In addition, we constrain annealing speed with a bounded cost function.

For bounded errors, as the range of the observed cost increases relative to the true cost, equilibrium properties and annealing speed diverge exponentially from true equilibrium and sequential annealing speed. For Gaussian errors, as the difference between the lowest and highest variances increases, equilibrium properties diverge exponentially. Finally, all these quantities diverge exponentially with the inverse-temperature.

Researchers have speculated that tuning errors to a constant factor of temperature helps the system converge. Our final result shows that guess to be correct, at least in the case of inverse logarithmic temperature schedules.

This work was supported by the IBM T.J. Watson Research Center. I gratefully acknowledge the comments of Frederica Darema, John Tsitsiklis, Albert Boulanger, and Griff Bilbro.

REFERENCES

[1] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[2] Philip N. Strenski and Scott Kirkpatrick. Analysis of finite length annealing schedules. Research Report RC 14672, IBM, June 1989.

[3] Daniel R. Greening. Parallel simulated annealing techniques. *Physica D: Nonlinear Phenomena*, 42:293–306, 1990.

[4] Daniel R. Greening and Frederica Darema. Rectangular spatial decomposition methods for parallel simulated annealing. In *Proceedings of the International Conference on Supercomputing*, pages 295–302, Crete, Greece, June 1989.

[5] Frederica Darema, Scott Kirkpatrick, and Alan V. Norton. Parallel algorithms for chip placement by simulated annealing. *IBM Journal of Research and Development*, 31(3):391–402, May 1987.

[6] Lov K. Grover. Simulated annealing using approximate calculation. In *Progress in Computer Aided VLSI Design, volume 6.* Ablex Publishing Corp., 1989.

[7] M.D. Durand and Steve R. White. Permissible error in parallel simulated annealing. Research Report RC 15487, IBM, 1990.

[8] Saul B. Gelfand and Sanjoy K. Mitter. Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications*, 62(1):49–62, July 1989.

[9] Prithviraj Banerjee, Mark Howard Jones, and Jeff S. Sargent. Parallel simulated annealing algorithms for cell placement on hypercube multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, 1(1):91–106, January 1990.

[10] Andrea Casotto, Fabio Romeo, and Alberto Sangiovanni-Vincentelli. A parallel simulated annealing algorithm for the placement of macro-cells. *IEEE Transactions on Computer-Aided Design*, CAD-6(5):838–847, September 1987.

[11] Daniel R. Greening. Simulated annealing with inaccurate cost functions. Technical report, UCLA Computer Science Dept., Los Angeles, 1991.

[12] Andrei Broder and Eli Shamir. On the second eigenvalue of random regular graphs. In *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science*, pages 286–294, 1987.

[13] Gregory B. Sorkin. Simulated annealing on fractals: Theoretical analysis and relevance for combinatorial optimization. In William J. Dally, editor, *Advanced Research in VLSI*, pages 331–351. MIT Press, Cambridge, Massachusetts, 1990.

[14] R.H.J.M. Otten and L.P.P.P. van Ginneken. *The Annealing Algorithm.* Kluwer Academic Publishers, Boston, 1989.

[15] Bruce Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, May 1988.

[16] John N. Tsitsiklis. Markov chains with rare transitions and simulated annealing. *Mathematics of Operations Research*, 14(1):70–90, February 1989.

Time-homogeneous Parallel Annealing Algorithm

Kouichi Kimura Kazuo Taki
Institute for New Generation Computer Technology
1-4-28 Mita, Minato-ku, Tokyo 108, Japan

Abstract

We propose a new parallel simulated annealing algorithm. Each processor maintains one solution and performs the annealing process concurrently at a *constant* temperature that differs from processor to processor, and the solutions obtained by the processors are exchanged occasionally in some probabilistic way. An appropriate cooling schedule is automatically constructed from the set of temperatures that are assigned to the processors. Thus we can avoid the task of carefully reducing the temperature according to the time, which is essential for the performance of the conventional sequential algorithm.

In this paper we propose a scheme of the probabilistic exchange of solutions and justify it from the viewpoint of probability theory. We have applied our algorithm to a graph-partitioning problem. Results of experiments, and comparison with those of the sequential annealing algorithm and the Kernighan-Lin algorithm, are discussed.

1 Introduction

Simulated annealing is a general and powerful technique to solve difficult combinatorial optimization problems [1]. It consists of many iterative *steps*. each modifies the current solution randomly and accepts it with probability $\min\{1,\exp(-\Delta E/T)\}$. Here $-\Delta E$ represents the gain obtained by the proposed modification in terms of the *energy* (objective function) E, and $T > 0$ is the *temperature* which is gradually reduced according to a *cooling schedule*.

Unfortunately, the theoretically optimal cooling schedule, which guarantees the convergence to the optimal solution, proves to be too slow for practical use [2]. Cooling schedules with geometrically decreasing temperatures are often used in applications. To obtain more elaborate cooling schedules is an active area of research [3].

In this paper we propose a new parallel simulated annealing algorithm. It automatically constructs an appropriate cooling schedule from a given set of temperatures.

2 An Annealing Algorithm Parallelized in Temperature

2.1 Outline of the Algorithm

The basic idea is to use parallelism for various temperatures, to perform annealing processes concurrently at various temperatures instead of sequentially reducing the temperature according to the time.

The outline of the algorithm is as follows. Each processor maintains one solution and performs the annealing process concurrently at a *constant* temperature that differs from processor to processor. After every k annealing steps, every pair of the solutions from the processors with adjacent temperatures are exchanged with some probability p, which is distinct for each pair. The algorithm can be stopped after any large number of steps and we will find a well-optimized solution on the processor that has the lowest temperature. We refer to $f = 1/k$ as the *frequency of (probabilistic) exchanges* and p as the *probability of exchange*.

Since exchanging the solutions between processors with different temperatures is nothing but changing the temperature for each participant solution, each solution will select its appropriate cooling schedule dynamically through successive competitions with others for lower temperature. However, since the temperature on each processor remains constant, the algorithm itself is *time-homogeneous*. Thus we can avoid the task of carefully reducing the temperature according to the time, which is essential for the performance of the conventional sequential annealing algorithm. In other words, this algorithm automatically decides how many steps should be taken at each temperature. the majority of steps should be devoted to some *critical* temperatures.

However, it is necessary to allocate an appropriate temperature to each processor beforehand. Namely, we have to specify a set of temperatures, from which the algorithm will construct a cooling schedule. This set should be chosen wisely according to the estimation of the equilibrium (static) relation between the temperature and the energy. It must cover the region of temperature, only in which the equilibrium energy varies virtually. Here the concepts of the *scales* by S. White will be useful [4] .

2.2 Probability of Exchange

Investigating the necessary condition which the probabilistic exchange must satisfy, we determine the probability of exchange.

Imagine that the annealing process is performed *independently* at each processor at a distinct constant temperature. Then the distribution of the solution in each processor approaches Boltzmann distribution of the respective temperature [3]. The lower the temperature is, the better the solution that will be found, but after a longer time.

Now we introduce probabilistic exchanges of the solutions between the processors and intend to accelerate the convergence of the solutions so that we can find a better solution at the lowest temperature more quickly.

Let $p(T,E,T',E')$ denote the probability of the exchange between two solutions with energy E and E', at temperatures T and T'. Since we expect a better solution at a lower temperature, we define $p(T,E,T',E') = 1$ if $(T-T')(E-E') < 0$.

On the other hand, if $(T-T')(E-E') \geq 0$, $p(T,E,T',E')$ is *uniquely* determined as follows. In order to accelerate the convergence, a probabilistic exchange of the solutions must not disturb the equilibrium distribution. Hence the detailed balance equation must hold between the distributions before and after the exchange:

$$\begin{aligned}\frac{1}{Z(T)}\exp(-\frac{E}{T})\cdot\frac{1}{Z(T')}\exp(-\frac{E'}{T'})\cdot p(T,E,T',E')\\ = \frac{1}{Z(T)}\exp(-\frac{E'}{T})\cdot\frac{1}{Z(T')}\exp(-\frac{E}{T'})\cdot 1\end{aligned}$$

where $Z(T)$ denotes the partition function. Therefore we obtain

$$p(T,E,T',E') = \begin{cases} 1 & \text{if } \Delta T\cdot\Delta E < 0 \\ \exp(-\frac{\Delta T\cdot\Delta E}{TT'}) & \text{otherwise}\end{cases}$$

$$\text{where} \qquad \Delta T = T - T', \qquad \Delta E = E - E'$$

Note that $p(T,E,T',E') > 0$ for $\forall T\ \forall E\ \forall T'\ \forall E'$. This means that a solution can go through a *non-monotonic* cooling schedule.

This probability is quite different from that of choosing a solution-temperature pair in the systolic statistical cooling algorithm by E. Aarts *et al.* [5]. The advantage of the former is that it does not contain the partition function and hence can be computed easily.

2.3 Monotonic Convergence Property

We verify that each probabilistic exchange of solutions in fact accelerates the convergence of the algorithm.

Let p denote the distribution of the solutions at an arbitrary time. It will change into pA after one annealing step at each processor, or into pC after one probabilistic exchange for each pair of solutions, where A and C are the respective transition probability matrices [6]. Let π denote the equilibrium distribution. It can be shown [6] that

$$D(\pi\|p) \geq D(\pi\|pA) \quad \text{and} \quad D(\pi\|p) \geq D(\pi\|pC)$$

where $D(\pi\|p)$ denotes Kullback-Leibler divergence of π and p, which represents the discrepancy between them [7]. Here strict inequalities hold unless $p = \pi$. Moreover $D(\pi\|p) \to 0$ follows from the observation in the subsequent subsection.

Hence the distribution of the solutions *monotonically* approaches the equilibrium distribution during the execution.

2.4 Time-homogeneity

The above algorithm is *time-homogeneous*: it has no control parameter to change over time. This has two implications.

Firstly, the behavior of the algorithm is described in terms of a time-homogeneous Markov chain. In general it is an irreducible and acyclic Markov chain over a finite state space. Hence we can easily establish its convergence property [6].

Secondly, in executing the algorithm, we can stop it at any time and examine whether a satisfiable solution has already been obtained. If one has not, we can resume it again for a better solution, and can just continue it as long as we like. In contrast, in the conventional simulated annealing it is necessary to re-schedule the temperature when we resume it, once it has entered the lowest temperature.

3 Experimental Results

We have implemented our algorithm for a graph-partitioning problem on the Multi-PSI/V2 [9], an MIMD parallel machine with 64 PEs.

(graph-partitioning problem) Given a graph $G = (\mathcal{V}, \mathcal{E})$, define a *label* on the vertices $\lambda : \mathcal{V} \to \{\pm 1\}$ so as to minimize E, where

$$E = - \sum_{(u,v)\in\mathcal{E}} \lambda(u)\lambda(v) + c \cdot (\sum_{v\in\mathcal{V}} \lambda(v))^2, \quad (c > 0\text{: constant})$$

This is an NP-hard problem. Kernighan-Lin algorithm efficiently gives its approximate solutions [8].

For a random graph G with 400 vertices and 2004 edges, we compare the results given by our algorithm with those by other methods [Fig.1].

(a) Time-homogeneous parallel annealing: All 63 processors performed 20,000 annealing steps each at distinct constant temperatures. The highest and lowest temperatures are determined empirically, and the other temperatures are determined so that adjacent ones have the same ratio. As for frequency of exchanges f, we examined various values ranging from 1/20,000 to 1/2. Each point represents the average over 30 runs with different sequences of random numbers.

(b) Sequential annealing: The cooling schedule consists of exactly the same sequence of 63 temperatures as above. 20,000 annealing steps are performed, which are divided equally between the 63 temperatures.

(c) Simple parallel annealing. Each of 63 processors executes the sequential annealing algorithm described in (b) using a distinct sequence of random numbers. The result is the best solution obtained by them.

(d) Kernighan-Lin: Kernighan-Lin algorithm is repeatedly applied several times until convergence.

We made the following observations from [Fig.1].

1) (a) gives the best solutions for a wide range of the frequency of exchanges: $1/1000 \leq f \leq 1/2$. Hence this algorithm is not sensitive to the value of f except for values that are too small. However a too large value of f incurs a high cost in exchanging the solutions between the processors. The execution time for $f = 1/100$ was less than 8% greater than that for $f = 1/1000$ [6].

2) Since 20,000 annealing steps are relatively small, (b) gives a worse solution than (d). However, in (a), the algorithm probabilistically selects an appropriate cooling schedule with 20,000 steps and gives a better solution.

3) Note that the total number of annealing steps in (a) and that in (c) are the same. (a) outperforms (c) unless f is too small.

4 Conclusion and Future Works

We have proposed the time-homogeneous parallel annealing algorithm, in which an appropriate cooling schedule is automatically and probabilistically constructed from a given set of temperatures.

The behavior of this algorithm is theoretically tractable, since it is described in terms of a *time-homogeneous* Markov chain. In particular we have proved its monotonic convergence property.

We have experimentally observed that this algorithm automatically constructed a better cooling schedule than that which assigned the same number of annealing steps at each temperature. We also observed that this algorithm is robust for the choice of the frequency of exchanges.

The following require further investigation.

(i) How many processors should we use?

(ii) How should we assign temperatures to the processors?

(iii) How do we find the *optimal* frequency of exchanges?

(iv) Does this algorithm probabilistically select the *theoretically best* cooling schedule, the *best* assignment of the annealing steps to each temperature?

5 Acknowledgments

We would like to thank N. Ichiyoshi, K. Rokusawa, and E. Sugino for valuable discussions.

References

[1] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecci, "Optimization by Simulated Annealing," *Science*, vol.220, no.4598 (1983).

[2] B. Hajek, "Cooling Schedule for Optimal Simulated Annealing," *Math. Oper. Res.*, 13 (1988).

[3] P.J.M. van Laarhoven, and E.H.L. Aarts, "Simulated Annealing: Theory and Applications", Reidel, (1987).

[4] S. R. White, "Concepts of Scales in Simulated Annealing," *Proc. IEEE ICCD* (1984).

[5] E. H. L. Aarts *et al.*, "Parallel Implementations of the Statistical Cooling Algorithm," *Integration*, 4, (1984).

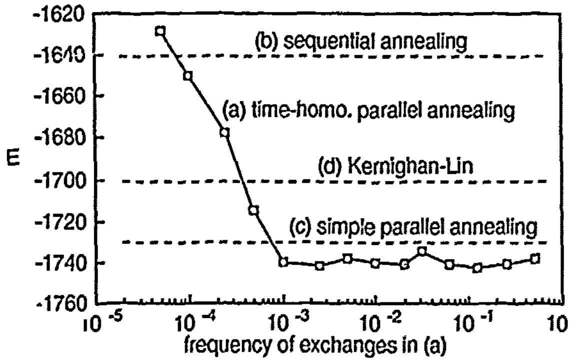
[6] K Kimura *et al.*, "On a Time-homogeneous Parallel Annealing Algorithm," *ICOT Technical Report*, 565 (*in Japanese*) (1990).

[7] S. Amari, "Differential Geometric Methods in Statistics," Lecture Note in Statistics 28, Springer-Verlag, (1985).

[8] B.W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell. sys. tech. J.*, 49, (1969).

[9] K Nakajima *et al.*, "Distributed Implementation of KL1 on the Multi PSI/V2", *Proc 6th Int. Conf. on Logic Programming* (1989).

Fig.1. energy vs frequency of exchanges



Parallelization of the Simulated Annealing Algorithm: Application to the Placement Problem

Bernard Virot (LIFO)
University of Orléans BP 6759
45067 Orléans Cédex 2 France

Abstract: We present a method of parallelization of the simulated annealing algorithm, applied to an instance of the chip placement problem. We give a mathematical evaluation for the synchronization cost and for the speedup of the method. We show that, for each stage of the algorithm, there exists an optimal number of processes, which depends only on a small number of measurable parameters. So, in order to obtain the best speedup, our method makes the number of employed processes vary dynamically during the execution of the algorithm.

An implementation of the method on a shared memory architecture is described, as well as its application to a real size problem: the placement of a graphic card made up of 272 chips and 638 equipotentials.

1. The Metropolis algorithm

Let us recall briefly the classical optimization method based on the Metropolis algorithm [1]. Let E be the energy function to minimize, defined on the state space Θ of a system . For each state in Θ, we define a set of neighboring states, and we call elementary move each transformation bringing a state x to a neighboring state y. We give a transition matrix $Q = (q_{xy})$ on $\Theta \times \Theta$, markovian, symmetric and irreducible, such that $q_{xy} > 0$ if and only if x is a neighbor of y and $x \neq y$.

We model the dynamic system by a Markov chain (X_n) over Θ, controlled by a parameter T called temperature, and defined in the following way [2]. Suppose $X_0, \dots, X_n$ have already been built. We choose at random an elementary move bringing to a state Y_n, according to the probability law

$$Proba(Y_n = y|X_0, \dots, X_n) = q_{X_n y}.$$

We impose, with probability 1,

$$(X_{n+1} = Y_n \; or \; X_{n+1} = X_n),$$

the choice being random according to the law

$$Proba(X_{n+1} = Y_n|X_0, \dots, X_n, Y_n) = \min\left(1, exp(-\frac{\Delta E}{T})\right).$$

Here ΔE denotes the energy variation corresponding to the elementary move bringing from X_n to Y_n. This move will be called a trial move. If $X_{n+1} = Y_n$, then we say it is an acceptable move. Otherwise $(X_{n+1} = X_n)$, it is a rejected move.

Now, if we take for T a decreasing function of n tending to 0 sufficiently slowly, ($T = T_n > \frac{C}{\log n}$, where C is a suitably chosen constant), then we can prove that the law of X_n converges to a measure supported by the set of the absolute minima of the energy function E (*cf.* [2]) .

2. Parallelization of the placement problem

In the sequel we study the parallelization of the Metropolis algorithm, applied to a specific type of problem, namely the chip placement problem. One can think of two dual parallelization methods:

- **Data partitioning:** One distributes the data (chip positions) among the processes.
- **Tasks partitioning:** One distributes the tasks among the processes, the data being shared.

With the data partitioning method a process can move only its own chips. In order to accept or reject a chip move, according to the Metropolis algorithm, one has to know the current length of all the equipotentials connected to that chip. But two chips belonging to two different processes can belong to the same equipotential. So, if erroneous computations of the energy variation are not admitted, then one must forbid simultaneous moves of such chips. Even for a small number of processes the synchronization time becomes prohibitive. For example, suppose that each equipotential connects 4 chips on the average, and each chip belongs to 10 equipotentials. Then, for every trial move of a chip, a process must lock those of the $4 \times 10 = 40$ chips which belong to other processes, i. e. 15 % of the total number in our case. If more than seven processes run simultaneously, then there exists an undesirable waiting time due to the lack of available chips. Let us point out that this method leads to a dynamic partitioning problem which can, in turn, be solved by simulated annealing [3].

With the tasks partitioning method the chip positions are shared data. One has to decide what exactly we mean by a task It may mean completing a move (coarse grain decomposition), or finding out an acceptable move (medium grain decomposition). One can even think of a fine grain decomposition where two or more processes cooperate for the computation of the energy variation involved by a single trial-move [4]. The coarse grain decomposition clearly requires a mutual exclusion protocol in order to avoid contradictory decisions. Moreover, while a chip is moved, all the chips belonging to the same equipotentials must be locked. The fine grain decomposition is "processor consuming" since each trial move involves the cooperation of several processes. It is suitable for a massively parallel architecture, where a processor can be efficient only for simple actions. The medium grain decomposition involves no lock at all, and only minimal synchronization. The parallelization method we studied corresponds to this choice. In the sequel, we will call it *the parallel trials method.*

3. The parallel trials method

The method was previously used by S. A. Kravitz, R. A. Rutenbar [5], and also by E. Aarts and J. Korst. Starting from a common initial state, the processes build independent Markov chains, until one of them (at least) obtains a number S, fixed in advance, of successive acceptable moves according to the Metropolis algorithm. Such a process will be called a winner. Then, all the processes synchronize. One chooses at random exactly one of the winners and lets its current configuration (resulting from its S acceptable moves) be communicated to the others processes. Then, one starts again from the new common state. Thus, each process executes the following algorithm.

```
while not Termination_test do
    found := false; my_count := 0;
    while (not found) do
        choose at random a move m;
        if Test(m) then
            my_count := my_count + 1;
            if (my_count >= S) then
                found = :TRUE;
            fi
        fi
    done
    Synchronize;
done
```

The variable my_count is private, whereas found is shared. Termination_test is the test defining the termination of the algorithm. In order to avoid deadlock, it must have the same value in all the processes. Test(m) is a function returning TRUE if the move m is acceptable and FALSE otherwise. In the procedure Synchronize the processes synchronize and agree on the starting configuration that they will use in the next step.

The role of the parameter S is crucial, because there are different phases in the execution of the algorithm. At low temperature, few of the trial moves are acceptable. So, synchronizing the processes for each acceptable move ($S = 1$) involves only a small waste of time, with respect to the synchronization time involved by a greater value of S. Moreover, choosing $S = 1$ maximizes the speedup produced by independent parallel searches of an acceptable move.

At high temperature the situation is quite different. Most of the moves are acceptable. So, the speedup produced by the parallel searches of an acceptable move is not very significant. If we synchronize the processes as soon as an acceptable move is found, then the synchronization time can become prohibitive. This will moreover favor the moves for which the decision is the fastest (for example, the translation of a chip with a small number of connections). In the opposite, if we choose S sufficiently large, then we obtain Markov chains long enough so that the total computation time of a chain can be considered independent of the choice of the trial moves. To sum up, at high temperature, the parameter S plays the following roles:

- reduce the synchronization overhead: th: greater is S, the smaller the number of process synchronization;

- make the acceptation probability independent of the types of the moves: for S sufficiently large, the computation time to find out a number S of acceptable moves can be considered independent of the complexities of the tried moves.

4. A small model

4.1 Local analysis (constant temperature)

In this section, we fix a low temperature T and we take $S = 1$. We denote by α the mean acceptation rate, that is the probability that a trial move be acceptable, following the Metropolis algorithm. If T is given, then α is constant. Let us call t_N the mean waiting time for the first acceptable move, when N processes make parallel independent trials. If α is sufficiently small, then we may estimate that

$$t_N = \frac{1}{1-(1-\frac{1}{t_1})^N} \approx \frac{t_1}{N}$$

Let us denote by s_N the mean time needed by N processes to synchronize and to agree on a new starting configuration. We denote by

$$r_N \approx t_N + s_N$$

the total mean time needed to achieve a move.

We assume s_N is proportional to the number of processes: $s_N \approx aN$ (we postpone the discussion of the validity of the hypothesis $s_N \approx aN$ until the end of the section 4), whence the formula

$$r_N \approx \frac{t_1}{N} + aN \qquad (1)$$

Note that, since t_1 is the mean waiting time for an acceptable move with exactly one process, we have

$$t_1 = \frac{c}{\alpha} \qquad (2)$$

where c is a constant, which can be viewed as the sum of the times to choose at random a move, to compute the corresponding variation of energy, and to take the decision of acceptation or rejection. The parameter c depends on the implementation of the algorithm, on the architecture, and on the instance of the placement problem.

An elementary computation shows that r_N, considered as a function of N, reaches its minimum $r_{N_{opt}}$ for

$$N = N_{opt} \approx \sqrt{\frac{t_1}{a}} = \sqrt{\frac{c}{a}}\sqrt{\frac{1}{\alpha}} \qquad (3)$$

and that we have

$$r_{N_{opt}} \approx 2\sqrt{at_1} \qquad (4)$$

Therefore, for a fixed temperature T, there exists an optimal number of processes $N = N_{opt}$. Moreover, N_{opt} is approximately proportional to $\alpha^{-\frac{1}{2}}$.

For this optimal value of the number of processes, the speedup G we obtain can be defined and computed in the following way:

$$G = \frac{r_1}{r_{N_{opt}}} \approx \frac{1}{2}(\sqrt{\frac{t_1}{a}} + \sqrt{\frac{a}{t_1}}) \qquad (5)$$

When the temperature is sufficiently low, the acceptance rate α is small and, therefore, t_1 is large. Thus, we may write

$$G \approx \frac{1}{2}\sqrt{\frac{t_1}{a}} = \frac{1}{2}N_{opt} \qquad (6)$$

Equation (3) above shows that G is then proportional to $\alpha^{-\frac{1}{2}}$

.

4.2 Global analysis (variable temperature)

We have chosen to decrease the temperature by steps, following the exponential law $T_k = (0,90)^k T_0$ where T_k denotes the k^{th} step's temperature.

In order to determine the length of the k^{th} step, one may consider two types of rules. Let L_0 denote the length of the Markov chain and L_1 the number of accepted moves, both counted from the beginning of the current step, and let C_0 and C_1 be suitably chosen constants (initially, $L_0 = L_1 = 0$, and we always have $L_0 \geq L_1$).

rule 1: The k^{th} step ends when $L_0 \geq C_0$ or $L_1 \geq C_1$;

rule 2: The k^{th} step ends when $L_1 \geq C_1$;

Let us remark that, in order to compute L_0, one has to decide how to take into account the moves rejected by the parallel processes. An extensive discussion of this problem can be found in [6].

At low temperature, under the first rule, the steps are cut down into a constant length determined by C_0, leading to an exponential decrease of the temperature. For our instance of the placement problem the experiments done with this rule highlighted the trap of the annealing in a local minimum. The second rule makes the steps longer, proportional to α^{-1}, where α denotes the current acceptance rate, slowing down the temperature decrease. Our experiments done with this second rule led to a mean energy 20 % smaller than with the first rule. In the sequel we therefore consider only the second rule. The need for a very slow temperature decrease seems mainly due to the constraints involved in our problem. We postpone the discussion of this point until section 5.

Let us denote by α_k the mean acceptance rate in step k, by $t_{N,k}$ the corresponding mean waiting time for the first acceptable move with N processes in parallel, and by $r_{N,k}$ the mean time needed to achieve a move. Equation (5) above shows that the optimal speedup G_k for the k^{th} step is then given by the relation

$$G_k \approx \frac{1}{2}\left(\sqrt{\frac{t_{1,k}}{a}} + \sqrt{\frac{a}{t_{1,k}}}\right)$$

or, since $t_{1,k} = \dfrac{c}{\alpha_k}$,

$$G_k \approx \frac{1}{2}\sqrt{\frac{c}{a\,\alpha_k}}\,(1 + \frac{a}{c}\alpha_k)$$

Therefore, if we choose dynamically, for every step, the optimal number N_{opt} of processes given by equation (3), then we have

$$r_{N,k} = \frac{1}{G_k} r_{1,k} = \frac{1}{G_k}(\frac{c}{\alpha_k} + a)\ ,$$

whence, finally the formula

$$r_{N,k} \approx 2\sqrt{\frac{ac}{\alpha_k}} \qquad (7)$$

Since all the steps contain the same number of accepted moves we may define the global speedup G_{glob} for K successive steps by the relation

$$G_{glob} = \frac{\sum_{k=1}^{K} r_{1,k}}{\sum_{k=1}^{K} r_{N,k}}$$

Using the equation (7) above, we can write G_{glob} as a function of the sequence of acceptation rates α_k, $1 \leq k \leq K$.

$$G_{glob} \approx \frac{1}{2}\sqrt{\frac{c}{a}}\ \left(\sum_{k=1}^{K}(\frac{1}{\alpha_k} + \frac{a}{c})\right)\left(\sum_{k=1}^{K}\frac{1}{\sqrt{\alpha_k}}\right)^{-1} \qquad (8)$$

Equation (8) allows the effective computation of G_{glob}, since a, c and α_k are easily measurable.

4.3 Discussion

In this section we discuss the relations of our method with the architecture and the implementation.

The algorithm contains a random choice among the winners, after every S accepted moves. On a shared memory architecture this choice can be made simply by a specialized process (master). On a distributed architecture it would probably be better to use a more symmetric algorithm, the winners electing one of them by means of an election algorithm.

We made the assumption that the synchronization and updating mean time s_N is proportional to the number of processes. Let us remark that s_N depends on the implementation of the algorithm and on the architecture, but it does not depend on the instance of the placement problem. On a shared memory architecture, if the synchronization protocol uses a critical section, then s_N is bounded below by a constant multiplied by the number N of processes. So, our hypothesis corresponds to a good implementation of a protocol belonging to this class. On a distributed architecture the situation can be quite different. For example, on a grid of N processors, the time to broadcast a value, and thus to update the data structures, is proportional to $\sqrt{N}$, and no critical section is needed. However, the corresponding proportionality constant is likely to be much greater than that of the shared memory case.

5. The placement problem

The previous algorithm is implemented on a Sequent machine (*Balance 8000*). It is a 32 bits shared memory multiprocessor architecture, with a single bus and a cache memory. The machine we used had eight processors. We studied one instance of the chip placement problem, namely the placement of a graphic card made of 272 chips and 638 equipotentials. The energy function we had to minimize has the following form

$$E = E_f + pE_r + qE_d$$

where we denote by

E_f the total length of the wires,

E_r the sum of the overlapping areas of the chips taken two by two,

E_d the sum of the areas of the chips parts extending beyond the limits of the board,

p and q two adjustable weights.

We have chosen the following elementary moves: the exchange of two chips, the rotations of angle $k\frac{\pi}{2}$, the translations small enough to ensure that a chip is never entirely outside the board. One imposes several constraints:

- Some chips, such as the connectors, must remain fixed.
- There are forbidden areas, that no chip may intersect
- When the annealing ends, we want to get $E_r = E_d = 0$: all the chips must fit in the rectangle formed by the board, without any overlap.

In order to satisfy (asymptotically) the latest constraint p and q must tend to $+\infty$ with the length n of the Markov chain, from the beginning of the simulated annealing. Let us denote by p_n, q_n and T_n the current values of the weights p, q, and of the temperature T. D. Geman and S. Geman [7] showed that the ratios $\frac{p_n}{T_n}$ and $\frac{q_n}{T_n}$ must tend to $+\infty$ very slowly, namely

$$\frac{p_n}{T_n} \leq C \log n \quad \text{and} \quad \frac{q_n}{T_n} \leq C \log n \,,$$

where C is a suitable constant. In particular, both the decrease of T_n and the growth of the weights p_n and q_n must be very slow. If we take $T_n = \frac{C}{\log n}$, then the above constraint on $\frac{p_n}{T_n}$ would imply $p_n \leq C^2$ and, therefore, the condition $p_n \to +\infty$ is not satisfied. This explains why temperature steps with constant length are not suitable for optimization problems with constraints (*see section 4.2 above*).

6. Numerical values

6.1 Local analysis

We determined experimentally the parameters we introduced in the section 4, by computing a mean on 5000 iterations at a constant temperature.

6.1.1 Synchronization and updating mean time s_N

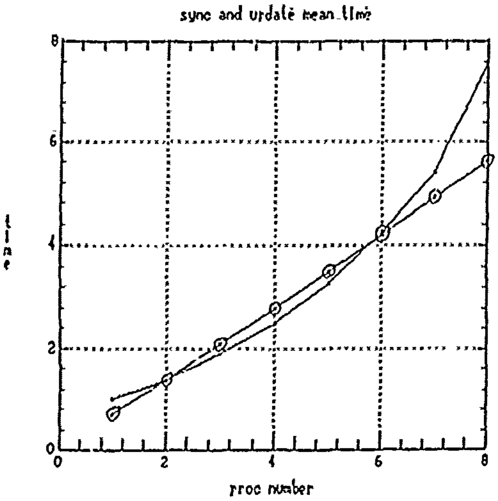


Fig. 1 *The theoretical graph is drawn with surrounded squares.*

We deduce from this graph (Fig. 1) the estimated value for a:

$$a \approx 0.7$$

The divergence of the theoretical graph from the experimental one for $N \geq 7$ is due to the processors'saturation. In fact, the hypothesis $s_N = a\,N$ is legitimate only if each process gets a processor as soon as it is ready to perform a step. This is not true when the number of processes is close to the number, of existing processors (eight), because in the *UNIX* system, at least one of the processors has to run the kernel's processes.

6.1.2 Move achievement mean time r_N

- $\alpha = 0.45$, $S = 1$. The mean acceptance rate is $\alpha = 0.45$, corresponding to a relatively high temperature, and we synchronize the processes as soon as an acceptable move is found.

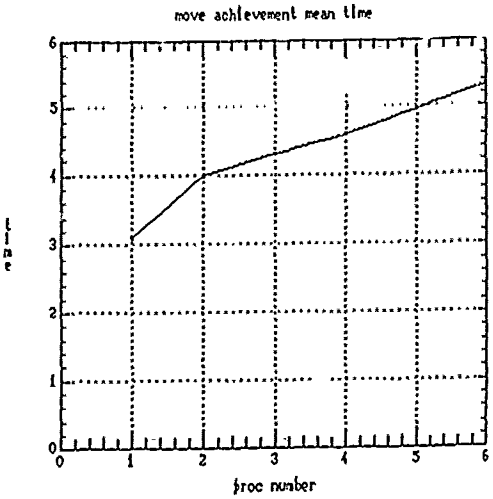


Fig. 2 $\alpha = 0.45$, $S = 1$

We can thus see that choosing $S = 1$ at high temperature slows down the algorithm (Fig. 2).

- $\alpha = 0.45$, $S = 4$. The mean acceptance rate is $\alpha = 0.45$, and we synchronize when a process has found 4 successive acceptable moves.

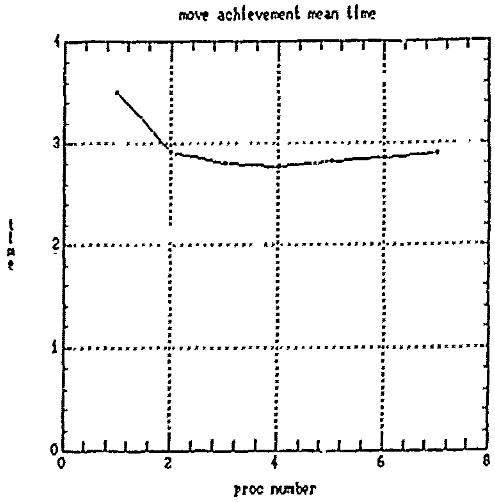


Fig. 3 $\alpha = 0.45$, $S = 4$

The local speedup is optimal when $N = 4$. Its measured value is $G = \frac{r_1}{r_4} = \frac{3.5}{2.78} \approx 1.2$ (Fig. 3).

- $\alpha = 0.05$, $S = 1$. The mean acceptance rate is $\alpha = 0.05$, corresponding to a low temperature, and we synchronize as soon as a process has found an acceptable move.

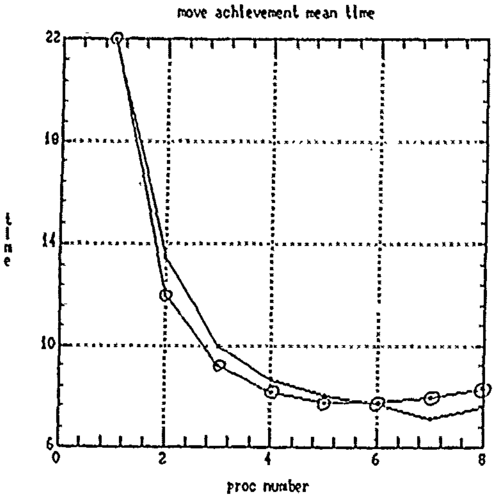


Fig. 4 $\alpha = 0.05$, $S = 1$. *The theoretical graph ($r_1 = 22$, $a = 0.7$) is drawn with surrounded squares*

In this case (low temperature) the experimental value of N_{opt} is 7 (Fig. 4). Its value computed by the relation (3) above, with $r_1 = 22$ et $a = 0.7$ is

$$N_{opt} = \sqrt{\frac{22 - 0.7}{0.7}} \approx 5.5$$

The measured speedup is $G = \frac{r_1}{r_7} = \frac{22}{7.1} \approx 3$ Its value computed by the relation (6) is

$$G = \frac{1}{2} N_{opt} \approx 2.7$$

6.1.3 Local speedup

With the notations of the section 4 above, the parameter a depends on the implementation of the algorithm and on the architecture. The values of t_1 and c depend also on the instance of the placement problem. Using their measured values and the relations (2), (4) et (6) above, one can compute the following table, which shows the estimated optimal number of processes and the local speedup, for different values of the acceptance rate α.

α	N_{opt}	G
0.05	6	3
0.01	13	6.5
0.005	18	9

6.2 Global analysis

In this section we study the algorithm behavior when the temperature is varying.

6.2.1 Global speedup

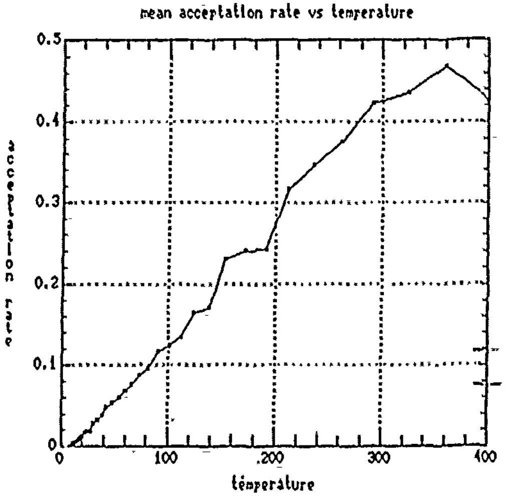


Fig. 5 *mean acceptance rate*

The graph (Fig. 5) above shows the measured variations of the mean acceptance rate considered as a function of the temperature. Using its values and the relation (8), one can compute the global optimal speedup G_{glob}, obtained when the algorithm runs, for each temperature, the optimal number of processes. For the instance of the placement problem we studied one obtains the value

$$G_{glob} \approx 7.5$$

The next graph (Fig. 6) shows the estimated optimal number of processes, computed by the relation (3). Recall that the machine we used had 8 processors. Thus, one see that we could not run this optimal number of processes for temperatures lower than 30.

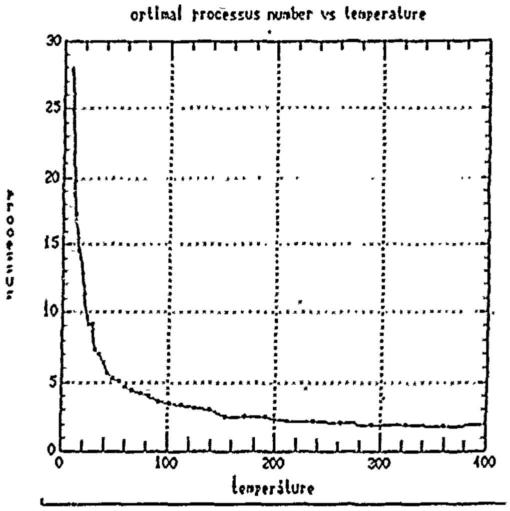


Fig. 6 *optimal processus number*

6.2.2 Energy

We performed five simulated annealing for the same instance of the placement problem . We obtained a mean final energy $m = 41177$ with a standard deviation $\sigma = 1011$. Let us remark that the initial placement of the card was done "*by hand*" in about a week, resulting in a final energy 30 % higher.

The following graph (Fig. 7) gives a typical example of the variation of the mean energy *versus* the temperature.

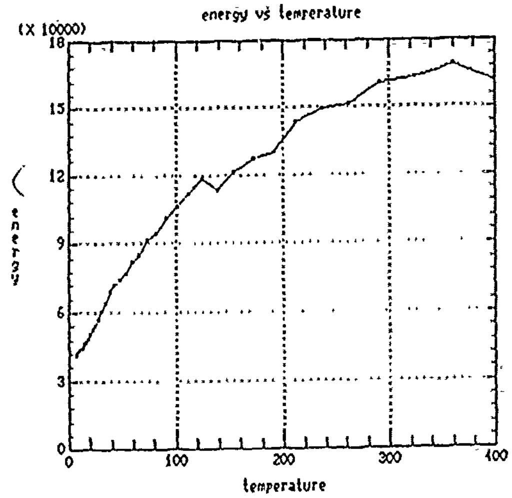


Fig. 7 *energy*

6.3 Feasibility

The card's description file, in a standard format, is read by the program through a parser. After the placement is completed, the program produces a file in the same format, describing the final card. This last file is then processed automatically by a routing software, demonstrating the feasibility of our placement solution.

Conclusion

The results above show that the parallelization method we studied is not a massive parallelization method. But, it gives a significant speedup at low temperature.

In order to obtain the best speedup, one must adapt dynamically the number of processes to the current mean acceptance rate. The notion of adaptative strategy was introduced by S. A. Kravitz and R. A. Rutenbar. In [5] they juxtapose different parallelization methods, leading to complex and, *a priori*, not optimal schedules. Our theoretical model allows us to compute the optimal number of processes, and the speedup, for each step of the algorithm.

The algorithm and the theoretical model are general enough to handle both shared memory and distributed architectures. They were tested only on a shared memory computer. However, on a distributed architecture, the synchronization and updating mean time, as a function of the number of processes, is likely to behave differently, leading to new estimates for the optimal number of processes and for the speedup. We plan to experiment on a distributed computer based on Transputers.

Acknowledgements.

It is a pleasure to acknowledge the help of Jean Jacques Levy at I.N.R.I.A., who put at our disposal a Sequent machine.

We also thank Philippe Matherat for the file describing his graphic card, and Marcin Skubiszewski who wrote the parser reading the file.

The present work was partially supported by the French GRECO C3.

BIBLIOGRAPHY

[1] S. Kirkpatrick, C.D. Gelatt Jr, M. P. Vecchi, Optimization by Simulated Annealing, Science, Vol. 220, 1983, p. 671–680.

[2] R. Azencott, Simulated Annealing, Séminaire Bourbaki, Exp. 697, 1988.

[3] A. Casotto, F. Romeo, A. Sangiovanni-Vincentelli, A Parallel Simulated Annealing Algorithm for the Placement of Macro-cells, Proc. IEEE Int. Conference on Computer Aided Design, Santa-Clara, 1986, p. 30–33.

[4] J. S. Rose, D. R. Blythe, W. M. Snelgrove and Z. G. Vranesic, Fast High Quality VLSI Placement on an MIMD Multiprocessor, Proc. IEEE Int. Conference on Computer Aided Design, Santa Clara, 1986, p. 42–45.

[5] S. A. Kravitz and R. A. Rutenbar, Placement by Simulated Annealing on a Multiprocessor, IEEE Trans. Computer-Aided Design, Vol. 6, 1987, p. 534–549

[6] P. Roussel-Ragot, G. Dreyfus, Etude de Différentes Méthodes de Parallélisation de l'Algorithme du Recuit Simulé: Modélisation et Expériences sur Réseau de Transputers (pre print).

[7] D. Geman, S. Geman, Relaxation and Annealing with constraints, Complex Systems Technical Report 35, 1987, Division of Applied Mathematics, Brown University, Providence, Rhode Island.

THREE-DIMENSIONAL ELECTROMAGNETIC PARTICLE-IN-CELL SIMULATIONS OF PHYSICAL DEVICES[1]

ALAN MANKOFSKY[2]
Applied Physics Operation
Science Applications International Corporation
1710 Goodridge Drive
McLean, Virginia 22012 USA

Abstract - We present an overview of three-dimensional electromagnetic particle-in-cell (PIC) simulation techniques for vector supercomputers, and their application to the realistic design of physical devices. We first desribe the fundamental building blocks of electromagnetic PIC codes. We then discuss code architectures for combining these components into a working design tool, using the ARGUS system of codes as a specific case study. Finally, applications and examples are discussed

I. INTRODUCTION

For more than twenty years, computers have been used to simulate systems of charged particles subject to both applied electromagnetic fields and to self-consistent fields generated by the particles themselves. Codes of increasing generality and complexity have evolved over this period. Today, with the widespread availability of vector supercomputers, three-dimensional codes which can self-consistently treat relativistic particles, electromagnetic fields, complex multimaterial structures embedded on the computational mesh, and realistic boundary conditions have become essential tools in the physics and engineering communities. The generality of such codes permits state-of-the-art modeling of, for example, microwave tubes (klystrons, magnetrons, traveling-wave tubes), accelerators, electron guns and electron optics systems, solid-state devices, and antennas. In effect, supercomputers have transformed electromagnetic PIC codes from research projects into truly cost-effective design and problem solving tools.

II. PHYSICS MODELS

The basic building blocks of an electromagnetic PIC code include the following:

Field solvers - These consist of both direct and iterative algorithms for solving particular subsets of Maxwell's equations. The full electromagnetic set and the electrostatic limit are the most common, with the magnetostatic and magnetoinductive approximations useful under certain circumstances. In addition, the full set of Maxwell's equations can be solved either in the time domain (as an initial value problem) or in the frequency domain (as an eigenvalue problem). The latter technique is an efficient and accurate way of determining electro magnetic spectra.

Particle pushers - These perform temporal orbit integration of relativistic particle species, and accumulate the source terms needed by the field solver. The particle equations of motion may include elastic and inelastic scattering from background species, interactions with matter, and terms describing rate processes.

"Kitchen sink" physics - Algorithms for describing surface physics, radiation, and other "dirty" processes are developed as needed. These can include phenomenological descriptions as well as more fundamental models.

Geometry specification and visualization - While not physics modules in the strict sense, these components are critical to a working three-dimensional PIC model. Complex geometries can be extremely difficult to set up in three dimensions, similarly, extracting some essential physical behavior from complicated three-dimensional field topologies and particle flows is often a daunting task. Combinatorial geometry tools (allowing for basic Boolean operations to be performed on three-dimensional objects) and advanced visualization packages (often workstation-based) are essential parts of an overall design system.

III. CODE ARCHITECTURE

Three-dimensional electromagnetic PIC models must handle vast quantities of data: a typical moderately-resolved simulation may require of the order of 100 million words of storage. Domain decomposition algorithms must be coupled with memory management and data management techniques for optimizing the use of fast memory for each calculation and for efficiently moving data between memory and disk as the calculation proceeds.

In SAIC's ARGUS code, for example, problems are decomposed into spatial regions known as ***field blocks***, which are then arranged on a set of generalized lattices known as ***logical supergrids***. The size of each field block is chosen so that the block will fit easily into memory; data-handling routines move blocks between disk and memory as needed. Disk I/O is overlapped with computations wherever possible. Algorithms used by both the field and particle routines permit global solutions in the entire physical domain by performing sequence-independent operations in each field block, followed by sharing of data at interfaces. Solutions in individual blocks may be advanced either synchronously or asynchronously; this architecture is naturally suited to parallel processing.

IV APPLICATIONS AND EXAMPLES

Three-dimensional electromagnetic PIC simulations are now widely used for research and design in a number of diverse areas. We briefly describe two different types of calculations that have been performed with the ARGUS code.

Microwave device and antenna design - The process of measuring the rf properties and mode structure of a microwave configuration with no applied voltages (*i.e.*, with no particles in it) is known as *cold testing*. It is typically the first series of measurements made when developing a new device. Once the cold-test properties of the structure are satisfactory, the designer will proceed to *hot testing*, or measurements where the device is under voltage and particles are present. As described above, cold testing can be performed in either the time domain or the frequency domain.

ARGUS has been used to predict the cold test behavior of numerous devices, including microwave cavities, high-power rf tubes, and high-frequency component packages. Typically, the code is first benchmarked against a configuration for which experimental data is available, agreement is usually found to within a fraction of a percent. Parametric scans are then performed to quantify the behavior of the device across the range of interest. It should be stressed that these devices are usually of sufficient geometrical complexity that analytical solutions are impossible (see Figures 1 and 2, for example). The only alternative to numerical simulation would then be to construct a prototype and perform an experimental study, a process which is usually far too costly and time-consuming.

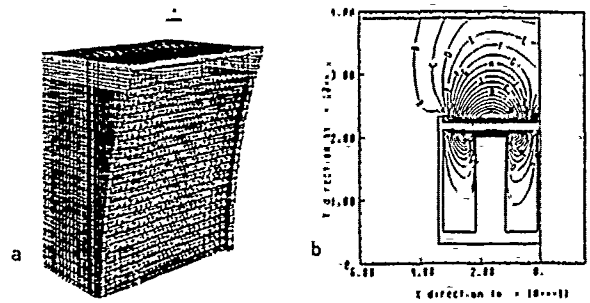


Fig 1 (a) ARGUS representation of a portion of a high-power antenna used for ion cyclotron resonance plasma heating in a Tokamak fusion reactor. (b) ARGUS generated contours of the z-component of the electric field at the midplane of the antenna when driven at its fundamental operating frequency.

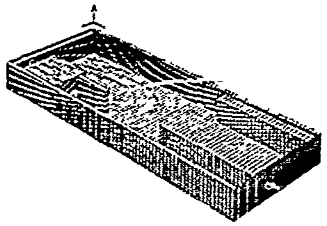


Fig. 2a ARGUS gridding of a microwave/millimeter wave integrated circuit module housing.

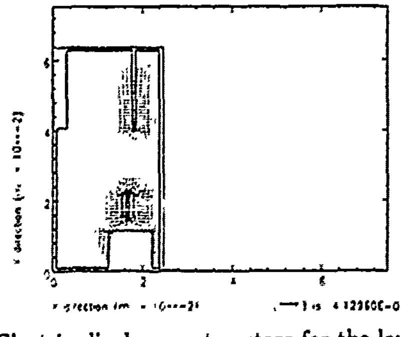


Fig. 2b Electric displacement vectors for the lowest mode of the device.

Accelerator design - The simulation codes traditionally used in the particle accelerator community have either employed simplified physics models (for example, ignoring space charge effects), or else have not been fully three-dimensional. However, modern accelerator designs are indeed three-dimensional, as can be seen from the Lawrence Berkeley Laboratory constant current variable voltage (CCVV) device shown in Figure 3(a). Fully three-dimensional PIC codes are therefore necessary to do a proper analysis.

We are currently using ARGUS to study such devices, with the goal of determining an optimal match between the accelerating system and attainable beam parameters. Preliminary results from such a calculation are shown in Figure 3(b).

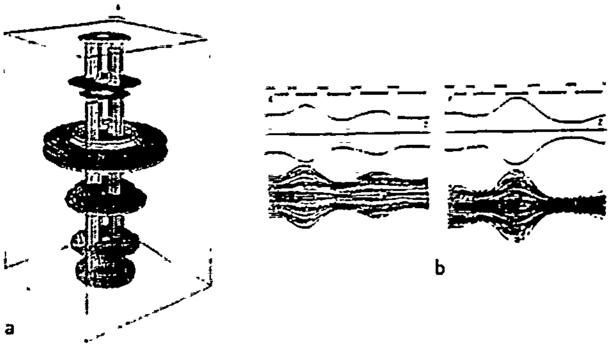


Fig. 3 (a) An ARGUS simulation of the LBL CCVV accelerator. The "fingers" are electrostatically charged in a quadrupole configuration so as to provide alternate gradient focusing and defocusing of the beam in the transverse plane as it is accelerated by the voltages on the plates. (b) Comparison between an ARGUS simulation and an envelope calculation for the CCVV accelerator. An x-z projection is on the left, and a y-z projection is on the right. The upper plots show the boundary of the beam as predicted by the envelope code, while the lower plots show the actual beam trajectories as calculated by ARGUS.

V. SUMMARY

Three-dimensional electromagnetic PIC simulation is a mature, well-developed, and cost-effective technique for the analysis of a wide variety of physical devices. As supercomputer-based codes become increasingly better coupled with workstation-based graphical capabilities, these integrated systems have moved to the forefront as everyday design tools. As always, work in the field is ongoing, and we expect to see even more dramatic advances over the next few years.

[1] Work supported by the SAIC Independent R&D Program and by the US Government.

[2] Also contributing to this work are O. Anderson, S. Brandon, C.F. Chan, C.L. Chang, A. Drobot, K. Ko, M. Kress, A. Mondelli, J. Moura, J. Petillo, and L. Soroka.

APPLICATION OF THE DETERMINISTIC PARTICLE METHOD TO THE WIGNER EQUATION

FRANCIS NIER
CMAP - Ecole Polytechnique
91128 Palaiseau Cedex
FRANCE

Abstract : The Wigner equation that we present here was proposed by physicists as a model for quantum electronic devices in the kinetic regime. The first member of this equation is a drift term while the quantum effects are taken into account in the second member via a Fourier integral operator. We solved this equation by a deterministic particle method for which proved the convergence.

I - The Wigner equation

The Wigner equation models the motion of electrons in an external electrostatic potential, which we shall decompose in order to describe quantum tunneling effects : the electrons are accelerated by an electric field E and partially tunnel through a potential barrier given by a real function V(x), $x \in \mathbb{R}^d$. This equation governs the evolution of a distribution function w(x,k,t), where $x \in \mathbb{R}^d$, $p \in \mathbb{R}^d$ and $t \in \mathbb{R}^+$ are respectively the position, the impulsion and the time coordinates :

(1.1)
$$\begin{cases} \partial_t w + \frac{p}{m}\partial_x w - qE\,\partial_p w = \Theta[V].w \\ w(t=0) = w_I \end{cases}$$

The real numbers $\hbar$, m and q are respectively the Planck constant, the mass of electron and its charge. The operator $\Theta[V]$ is a Fourier integral operator defined by

(1.2)
$$\left(\Theta[V].w\right)(x,p,t) = \frac{1}{(2\pi)^d}\iint_{\mathbb{R}^{2d}} \frac{1}{i\hbar}\left[V\left(x+\frac{h\eta}{2}\right) - V\left(x-\frac{h\eta}{2}\right)\right] \times$$
$$\times\, w(x,p',t)\, e^{i(p-p')\eta}\, dp'\, d\eta$$

This integral can also be written as a convolution with respect to the impulsion variable :

(1.3)
$$\left(\Theta[V].w\right)(x,p,t) = \int_{\mathbb{R}^d} \varphi(x,p-p')\, w(x,p',t)\, dp' \quad ,$$

with

(1.4)
$$\varphi(x,p) = \left(\frac{2}{h}\right)^{d+1}\left(\frac{1}{\sqrt{2\pi}}\right)^d \mathrm{Im}\left[e^{i\frac{2px}{h}}\,\hat{V}\left(\frac{2p}{h}\right)\right]$$

and

(1.5)
$$\hat{V}(u) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_{\mathbb{R}^d} V(x)\, e^{-iux}\, dx \quad .$$

A classical semi-group analysis provided in [1] states that the operator $-\frac{p}{m}\partial_x + qE\,\partial_p - \Theta[V]$ generates a continuous unitary group in $L^2(\mathbb{R}^{2d})$. Thus, if the initial data belongs to $L^2(\mathbb{R}^{2d})$, this equation admits a unique solution which satisfies

(1.6)
$$\| w(t) \|_{L^2} = \| w_I \|_{L^2} \quad .$$

II - The particle method

In order to solve numerically, we used the deterministic particle method introduced by P.A. Raviart and S. Mas-Gallic in [2] [3]. The exact solution of equation (1.1) is approximated by a linear combination of delta functions :

(2.1)
$$w(x,v,t) \approx \sum_i \omega_i\, w_i(t)\, \delta(x - x_i(t))\, \delta(p - p_i(t)) \quad .$$

The motion of the particles in the phase space is given by

(2.2)
$$\frac{dx_j}{dt}(t) = \frac{p_i(t)}{m} \quad , \quad \frac{dp_i}{dt}(t) = -qE \quad ,$$

while the control volume of the particle i, ω_i, does not depend on t and its weight, $w_i(t)$, is an approximation of $w(x_i(t), p_i(t), t)$. The weights evolve according to

(2.3)
$$\frac{dw_i}{dt}(t) = \sum_j \omega_j\, \zeta^\varepsilon(x_i(t) - x_j(t))\, \varphi(x_j(t), p_i(t) - p_j(t))\, w_j(t) \quad ,$$

that we obtain by first regularizing the integral operator in the x direction

$$\left(\Theta[V].w\right)(x,p,t) \approx \iint_{\mathbb{R}^{2d}} \zeta^\varepsilon(x-x')\,\varphi(x',p-p') \times$$
$$\times\, w(x',p',t)\, dp'\, dx' \quad ,$$

and then using a quadratural formula.

The numerical analysis of this method relies on $W^{m,p}$ estimates for the solution of equation (1.1). This estimates can not be obtained by the classical semi-group analysis which yields only (1.6). By using the convolution form of the integral operator, we have proved in [4] the estimate

(2.4)
$$\| w(t) \|_{W^{m,p}} \leq C(t) \| w_I \|_{W^{m,p}} \quad ,$$

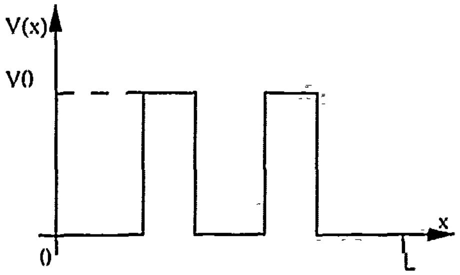
under the assumption

(2.5)
$$\operatorname*{Sup}_{|\alpha|+|\beta| \leq m} \left[\int_{R^d} \left| u^\alpha \partial^\beta \hat{V}(u) \right| du \right] < \infty \quad .$$

Then the classical arguments of consistence and stability yield the convergence of the method with order m

III - Application to the simulation of a resonant tunneling diode

A model proposed by physicists [5] for the simulation of resonant tunneling diodes relies on the one-dimensionnal (d=1) Wigner equation where the potential V(x) describe a double barrier.



The equation (1.1) has to be solved in a bounded domain $[0,L] \times [p_{min}, p_{max}]$ with physical boundary conditions given in [5] at the boundaries x=0 and x=L, and artificial boundary conditions for $p = p_{min}$ and $p = p_{max}$. We chose periodic boundary conditions which are the easiest one to take with the particle method. This model can be improved by coupling with the Poisson equation in order to take into account the electrostatic interaction. Then the electric field depends on the position $E = E(x)$ and is derived from the electrostatic potential. We compute this potential by a Particle In Cell method commonly used for this coupling [6] [7].

The main difficulty of this problem is due to the singularity of the barrier potential of which the Fourier transform does not satisfy the inequality (2.5) even for m=0. The particle method was proved to converge in this case but in a very weak sense and without any order. Moreover, the integral (1.3) is an oscillating integral which makes the numerical computations rather heavy. In spite of this, we got results which allowed to justify the simplified models used by physicists.

REFERENCES

[1] P.A. Markowich . "On the equivalence of the Schrödinger and the Quantum Liouville equations", Math. Math. in Applied Sci. 11 459-469 (1989).

[2] P.A. Raviart : "An Analysis of Particle Methods", Lecture Notes in Mathematics 1127.

[3] S.Mas-Gallic and P.A. Raviart : "A particle method for first order symmetric systems", Numer. Math. 51 323-3525 (1987).

[4] A. Arnold and F. Nier : "Numerical Analysis of the Deterministic Particle Method applied to the Wigner Equation", submitted.

[5] W.R. Frensley . "Wigner function model for a resonant-tunneling semi-conductor device", Phys. Rev. B 36 1570-1580 (1987).

[6] F. Guyot-Delaurens and P. Degond: "Particle Simulation of the Semiconductor Boltzmann equation for One -Dimensionnal Inhomogeneous Structures", Journal of Comp. Phys. 90 65-97 (1990).

[7] R.W. Hockney and J.W. Eastwood : Computer simulation using particles , Mc Graw-Hill (1981).

NUMERICAL INVESTIGATION OF THE STRONG EVAPORATION OF A POLYATOMIC GAS

Carlo Cercignani
Dipartimento di Matematica del Politecnico di Milano
Piazza Leonardo da Vinci 32 - 20133 - Milano - Italy

Aldo Frezzotti
Dipartimento di Matematica del Politecnico di Milano
Piazza Leonardo da Vinci 32 - 20133 - Milano - Italy

Abstract The aim of the present paper is the presentation of some results about the numerical study of the flow of a polyatomic gas between one evaporating and one totally absorbing plate. Density, velocity and temperature profiles are obtained in the range $0.01<Kn<0.1$ by solving a kinetic equation by a combination of Monte Carlo and finite difference techniques.

I. INTRODUCTION

The mathematical description of evaporation and condensation phenomena is a classical problem of kinetic theory and a large number of papers has been devoted to the study of its various aspects.

Most of the previous investigation were aimed at studying evaporation problems connected to monatomic gases. However, in many situations, which are relevant both from the scientific and practical point of view, one is faced with the evaporation of polyatomic species. The strong evaporation of polyatomic gas in a half space has been considered by Cercignani[1] who extended Ytrehus' trimodal Ansatz[2] to obtain approximate solutions of a BGK-like kinetic model by moment method. The aim of the work described in the present paper is the modeling of the evaporation of a polyatomic gas by the direct numerical solution of a kinetic equation. The approach has the obvious advantage of requiring no a priori guess of the distribution function, therefore it has a wider range of applicability as compared to the moment method. The one-dimensional flow resulting from the evaporation / condensation of a polyatomic gas between two parallel plates is considered as model problem.

II. BASIC EQUATIONS

As is well known the mathematical description of the behavior of a polyatomic gas is still an open problem that has been approached in a variety of different ways[3].

Holway's kinetic equation[4]

$$\frac{\partial f}{\partial t} + \underline{\xi}\nabla f = \nu_{el}(\Psi_{el} - f) + \nu_{an}(\Psi_{an} - f) \qquad (1)$$

was used as a starting point for this study, since it lends itself to a simple numerical treatment. In Eq. (1) the functions Ψ_{el} and Ψ_{an} are defined as follows:

$$\Psi_{el} = \frac{n(\varepsilon)}{(2\pi RT_t)^{3/2}} \exp\left[- \frac{(\underline{\xi} - \underline{U})^2}{2RT_t} \right] \qquad (2)$$

and

$$\Psi_{an} = \frac{N}{(2\pi RT)^{3/2}} \exp\left[- \frac{(\underline{\xi} - \underline{U})^2}{2RT} \right] \times$$

$$\frac{\varepsilon^{J/2-1}}{\Gamma(J/2)(kT)^{J/2}} \exp\left(- \frac{\varepsilon}{kT} \right) \qquad (3)$$

Eq. (1) approximates the collisional term by the sum of two BGK-like terms. The first of them describes elastic collisions, while the second models anelastic collisions. In Eq. (1) $f(x,t,\underline{\xi},\varepsilon)$ is the distribution function of molecular velocities $\underline{\xi}$ and internal energy ε of a gas with J internal (indistinguishable) degrees of freedom. The quantity $n(x,t,\varepsilon)$ is the number density of molecules having the internal energy ε, $N(x,t)$ is the total number density, $U(x,t)$ is the mean velocity of the gas, $T_t(x,t)$ is the translational temperature, $T(x,t)$ is the overall temperature. The collision frequencies have been calculated from the viscosity $\mu(T_t)$:

$$\nu_{el} = (1-z)\nu_{tot}, \quad \nu_{an} = z\nu_{tot}, \quad \nu_{tot} = NkT_t/\mu(T_t)$$

being z the fraction of anelastic collisions.

Since the main difficulty lies in the modeling of anelastic collision, it seems reasonable to restrict the use of BGK terms to that process only. A second model was therefore obtained by replacing the first collisional term by the full Boltzmann Equation to describe elastic collisions. The second term was left unchanged. If hard sphere interaction is assumed to describe elastic collisions, then Eq. (1) takes the following new form :

$$\frac{\partial f}{\partial t} + \underline{\xi}\,\nabla f =$$

$$(1-z)\frac{d^2}{2}\int [F(\underline{\xi}^*)f(\underline{\xi}^*,\varepsilon) - F(\underline{\xi}_1)f(\underline{\xi},\varepsilon)] \times$$

$$|\underline{\xi}_r \circ \hat{k}| \sin\theta \, d\underline{\xi}_1 d\theta \, d\phi + z\,\nu_{Coll}\,(\Psi_{an} - f) \qquad (4)$$

In Eq. (4) d is the molecular diameter and ν_{Coll} is the mean collision frequency calculated as :

$$\nu_{Coll} = \frac{1}{N}\,\frac{d^2}{2}\int F(x,t,\xi_1)F(x,t,\xi_2) \times$$

$$|\underline{\xi}_r \circ \hat{k}| \sin\theta \, d\underline{\xi}_1 d\underline{\xi}_2 d\theta \, d\phi \qquad (5)$$

The reduced distribution function $F(x,t,\xi)$ is defined as $F(x,t,\xi) = \int f(x,t,\underline{\xi},\varepsilon)d\varepsilon$.

The kinetic equations are solved specifying the initial values $f(x,0,\underline{\xi},\varepsilon)$ and the boundary conditions:

$$f(0,t,\underline{\xi},\varepsilon) = \Psi_{an}(N_w,T_w,\underline{\xi}) \qquad \xi_x > 0 \qquad (6)$$

$$f(L,t,\underline{\xi},\varepsilon) = 0 \qquad \xi_x < 0 \qquad (7)$$

The boundary condition (6), which holds at the evaporating plate, specifies the distribution function of the molecules emitted from the condensed phase. The temperature T_w is the wall temperature, while N_w denotes the saturated vapor density at the temperature T_w. The boundary condition (7) at x=L implies that the second plate is perfectly absorbing. As is clear from Eq. (6) it has been tacitly assumed that the evaporation coefficient is unit.

III. Description of the Numerical Technique

The numerical algorithm is based on a consistent finite difference discretization of Eqs. (1) and (4). The Monte Carlo method is used to evaluate the collision integral at the right hand side of Eq. (4). A generalization of the time-splitting method by Tcheremissine and Aristov[5] was used.

In this work, the region between the plates has been divided into a number of cells of variable size, and the distribution functions assumed to be constant within each cell. The size of spatial cells was smaller in the regions of stronger gradients. A similar procedure has been adopted to represent the distribution functions in the velocity space. A regular net of nodes is arranged into a finite domain of the velocity space assuming that the distribution function is constant within the cell volumes. The domain has to be chosen in order to contain most of the particles at any stage of the calculations. As far as the discretization of f in the space of internal energy ε is concerned, it was convenient to represent the distribution function through a set of values calculated at the nodes of a Gauss integration formula with unit weight function on the interval $[0,E_{max}]$. The distribution function was assumed to vanish for $\varepsilon>E_{max}$.

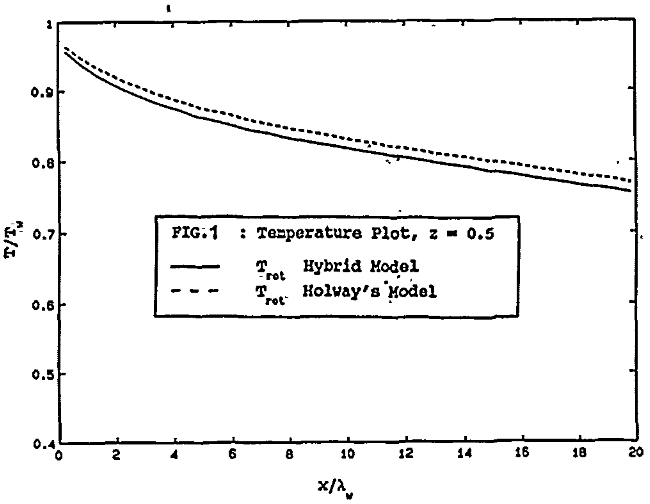
IV. RESULTS AND DISCUSSION

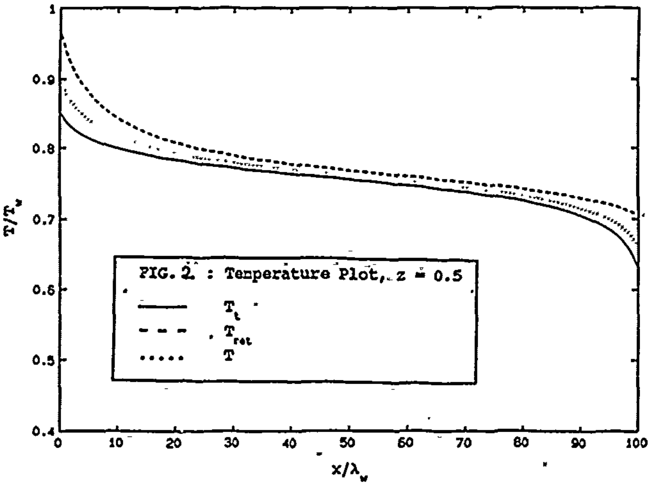
The numerical method described above has been used to obtain approximate solutions of Eq. (1) and Eq. (4) for values of the Knudsen number Kn in the range [0.01,0.1]. The Knudsen number considered here is defined as λ_w/L, being λ_w the mean free path when the gas is in equilibrium with the condensed phase. Only the case of a diatomic molecule (j=2) was considered. The values of the parameter z were selected in order to have the anelastic collision frequency not too different from the elastic collision frequency. In this way the relaxation phenomena associated with the internal and translational degrees of freedom occur on a scale of comparable magnitude.

The first result of the analysis was that Eq. (1) and Eq. (4) give very close results for the same values of Kn and z. The largest difference was found in the profiles of the rotational temperature shown in Fig. 1 in the case Kn=20, z=0.5. The difference is due to the fact that the value of the overall collision frequency given by Eq. (5) is slightly higher than the value calculated from viscosity. Therefore, most of the results have been obtained from Holway's model. The effects of the internal degrees of freedom are more evident in the translational temperature profiles (Fig. 2). The density and velocity profiles are very close to those of a monatomic gas. Finally, values of the back scatter fraction are presented in table 1. It was found that, as expected on the ground of the moment method calculations of Ref. 2, a larger fraction is scattered back to the surface in the case of a polyatomic gas.

Table 1 : Back scatter fraction vs. 1/Kn

z \ 1/Kn	10	20	50	100
0	0.130	0.143	0.155	0.160
0.5	0.142	0.167	0.183	0.188





V. REFERENCES

1. Cercignani, C.,"Strong Evaporation of a Polyatomic Gas". Rarefied Gas Dynamics, ed by S. S.Fisher, AIAA, N.Y.), 1, pp. 305-319 (1981).

2. Ytrehus, T.,"Theory and Experiments on Gas Kinetics in Evaporation". Rarefied Gas Dynamics, ed. by J. L. Potter, AIAA, N.Y., 2, pp. 1197-1212, (1977).

3. Bird, G.A.,"Molecular Gas Dynamics", Clarendon Press, Oxford , 1976.

4. Holway, L. H.,"New Statistical Models for Kinetic Theory". Phys. Fluids 9, pp. 1658-1673, 1966.

5. Aristov, V. V. and Tcheremissine, F.G.,"The conservative splitting method for solving the Boltzmann Equation". USSR Compt. Math. and Math.Phys. 20, pp. 208-225, 1980.

KINETIC RIEMANN SOLVERS FOR EULER EQUATIONS

B. PERTHAME
Département de Mathématiques
Université d'Orléans
BP 6759
45067 Orléans Cedex 2 (FRANCE)

Abstact : a family of riemann solvers for the Compressible Euler equations is presented, which preservs physical properties of the flow. They are obtained using a kinetic approach to the Euler Equations .

I - INTRODUCTION

We derive from the kinetic theory of gas, a family of Riemann type solvers for the Compressible Euler Equations of fluid dynamics that we call Boltzmann solvers.

The system we solve is

$$\partial_t \rho + \partial_x\ \rho u_1 + \partial_y\ \rho u_2 = 0\ ,$$
$$\partial_t\ \rho u_1 + \partial_x\ (\rho u_1^2 + p) + \partial_y\ \rho u_1\ u_2 = 0,$$
$$\partial_t\ \rho u_2 + \partial_x\ \rho u_1 u_2 + \partial_y\ (\rho\, u_2^2 + p) = 0$$
$$\partial_t\ E + \partial_x\ (E + p) u_1 + \partial_y\ (E + p) u_2 \quad = 0,$$
$$E = 1/2\ \rho\, |u|^2 + \rho\, e\ , \quad p = \rho T = (\gamma - 1)\rho e\,.$$

And we will use a rectangular mesh

$$M_{ij} = \{ (x,y) , | x - x_i | \leq \Delta x /2 , | y - y_j | \leq \Delta y /2 \}$$

where $x_i = i\ \Delta x$, $y_j = j\,\Delta y$. The problem we address is to find finite volume approximations of the above system under the form

$$U_{ij}^{n+1} - U_{ij}^{n} + \sigma_x (F_{i+1/2,j} - F_{i-1/2,j})$$
$$+ \sigma_y\ (G_{i,j+1/2} - G_{i,j-1/2}) = 0$$

$\sigma_x = \Delta t / \Delta x$, $\sigma_y = \Delta t / \Delta y$, where $F_{i+1/2,j}$ for example is an appropriate approximation of

$$(\rho u_1,\ \rho u_1^2 + p,\ \rho u_1 u_2,\ (E + p)\ u_1)\ (x_{i+1/2}, y_j)$$

such that , under a CFL condition,

(i) $\rho_{ij}^{n+1} \geq 0$, $T_{ij}^{n+1} \geq 0$,
(ii) the entropy inequality holds
(iii) the maximum principle on the entropy holds

$$S_{ij}^{n+1} \leq \ \mathrm{Max}\{ S_{k,\ell}^{\ n+1} ,\ M_{k,\ell} \text{ neighbor mesh to } M_{ij} \}$$

where S is an entropy of Euler Equations for instance

$$S = \rho^{(\gamma-1)}/T .$$

These properties can be realized using an exact solver (Godunov) i.e. by computing the exact value $U_{i+1/2,j}$ at the mesh interface.

II. BOLTZMANN SCHEMES.

We present another method, which allows to possibly take into account the corners effect (dependency of $F_{i+1/2,j}$ on $U_{i+1,j+1}$ for instance)

This is achieved using a kinetic approximation to Euler Equations

$$\partial_t\ f + v\,.\,\nabla f = 0 \qquad \text{on} \quad (n\,\Delta t, (n+1)\,\Delta t)$$
$$f\,(n\Delta t,z,v) = \rho^n/T^n\ \ \chi[(v - u^n)/\sqrt{T^n}\,]$$

where $\int (1, w_1^2,\ w_2^2)\ \ \chi(w)\, dw\ =\ (1, 1, 1)$ and $0 \leq \chi(w) \leq \chi(-w)$. The classical choice of $\chi\,(w)$ is the Maxwellian $\chi\,(w) =\ \exp\,(-|w|^2/2)\,/\,2\pi$ (see Deshpande for instance), In order to preserve (i) - (iii) we propose to use other choices of χ, which also give better numerical results.

Then, $F_{i+1/2,j}$ is <u>explicitely</u> and easily caltulated using at $z = (x_{i+1/2}, y_j)$ the exact solution to the transportequation

$$F_{i+1/2,j} = \int v_1 (1, v_1, v_2, | v |^2 /2 + \lambda T)^t\ \ f(n\Delta t, zv)\ dv\ ,$$
$$\lambda = .5\ (2 - \gamma)/(\gamma - 1).$$

Indeed polynomial choices of χ lead to integrate polynomes in order to get $F_{i+1/2,j}$ (See Perthame SIAM J. Num. Anal. 1991, to appear). The properties (i)-(iii) are proved for the numerical scheme for particular choices of χ , in noticing that they hold for the transport equation and that we solve exactly the transport equation.

Neglecting the corners, the explicit formula for $F_{i+1/2,j}$ is for the first order scheme

$$F_{i+1/2,j} = F^{+}(\,U_{i,j}\,) +\ F^{-}(\,U_{i+1,j}\,)$$

with

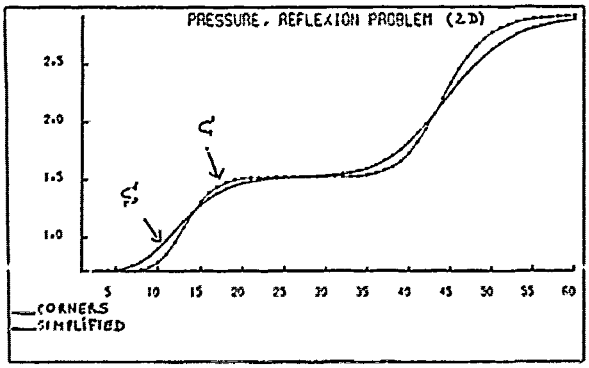
$$F^{+}(\,U\,) = \int_{v \geq 0}\ v\ \ (1, v, u_2, |v|^2/2 + |u_2|^2/2 + (1/2 + \lambda)T)^t$$
$$\xi[(v - u)/\sqrt{T}\,]\ \ dv,$$

and F^- is obtained integrating for $v \leq 0$. In this simplificated formula we have specified

$$\chi(w) = \xi(w_1)\ \xi(w_2) \text{ with}$$
$$\int (1, w^2)\ \ \xi(w)\, dw = (1,1)\ \ \text{and } 0 \leq \xi(w) \leq \xi(-w)\,.$$

III. NUMERICAL RESULTS.

The following figure presents, for a 2D stationary reflection problem, the improvment obtained using the exact solver, using corners, compared to the simplified solver described above.



A MACSYMA PROGRAM FOR THE HIROTA METHOD

WILLY HEREMAN AND WUNING ZHUANG
Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80401, USA

Abstract – Hirota's method for finding soliton solutions of nonlinear evolution and wave equations is briefly discussed and illustrated. A MACSYMA program that automatically carries out the lengthy algebraic computations is included.

I. INTRODUCTION

Hirota's direct method [1,2] allows to construct exact soliton solutions of nonlinear evolution and wave equations. The lengthy but straightforward calculations inherent to this method can easily be performed with any symbolic manipulation program. In this paper we present a sample program in MACSYMA to illustrate the symbolic calculation of one, two and three soliton solutions of well-known nonlinear PDEs such as the Korteweg-de Vries, the Boussinesq, the Kadomtsev-Petviashvili, the Sawada-Kotera and the shallow water wave equations [1-5].

II. THE HIROTA METHOD

Hirota's method requires:

(i) a clever change of dependent variable,

(ii) the introduction of a novel differential operator,

(iii) a perturbation expansion to solve the resulting bilinear equation.

Details about the method can be found in almost any book on soliton theory [1-4], here we merely outline the procedure.

Our leading example is the Korteweg-de Vries equation [1-4],

$$u_t + 6uu_x + u_{3x} = 0 \ . \tag{1}$$

Substitution of

$$u(x,t) = 2\frac{\partial^2 \ln f(x,t)}{\partial x^2} \tag{2}$$

into (1) and one integration with respect to x yields,

$$f f_{xt} - f_x f_t + f f_{4x} - 4 f_x f_{3x} + 3 f_{2x}^2 = 0 \ . \tag{3}$$

This quadratic equation in f can then be written in *bilinear form*,

$$B(f{\cdot}f) \overset{\text{def}}{=} \left(D_x D_t + D_x^4\right)(f{\cdot}f) = 0 \ . \tag{4}$$

where the new operator is given by

$$D_x^m D_t^n (f{\cdot}g) = (\partial x - \partial x')^m (\partial t - \partial t')^n f(x,t)\, g(x',t')\Big|_{x'=x,t'=t} \ . \tag{5}$$

Introducing a book keeping parameter ϵ, we look for a solution

$$f = 1 + \sum_{n=1}^{\infty} \epsilon^n f_n \ . \tag{6}$$

Substituting (6) into (4) and equating to zero the powers of ϵ, yields

$$O(\epsilon^0) \ : \ B(1{\cdot}1) = 0 \ , \tag{7}$$

$$O(\epsilon^1) \ : \ B(1{\cdot}f_1 + f_1{\cdot}1) = 0 \ , \tag{8}$$

$$O(\epsilon^2) \ : \ B(1{\cdot}f_2 + f_1{\cdot}f_1 + f_2{\cdot}1) = 0 \ , \tag{9}$$

$$O(\epsilon^3) \ : \ B(1{\cdot}f_3 + f_1{\cdot}f_2 + f_2{\cdot}f_1 + f_3{\cdot}1) = 0 \ , \tag{10}$$

$$O(\epsilon^4) \ : \ B(1\, f_4 + f_1{\cdot}f_3 + f_2\, f_2 + f_3{\cdot}f_1 + f_4{\cdot}1) = 0 \ , \tag{11}$$

$$O(\epsilon^n) \ : \ B(\sum_{j=0}^{n} f_j{\cdot}f_{n-j}) = 0 \ , \qquad \text{with } f_0 = 1 \ , \tag{12}$$

This scheme is general whatever the explicit expression of the bilinear operator B is. For the KdV equation B is given in (4).

If the original PDE admits a N-soliton solution then (6) will truncate at level $n = N$ provided f_1 is the sum of precisely N simple exponential terms. For simplicity, consider the case of a three soliton solution ($N = 3$). Then,

$$f_1 = \sum_{i=1}^{3} \exp(\theta_i) = \sum_{i=1}^{3} \exp(k_i x - \omega_i t + \delta_i) \ , \tag{13}$$

where k_i, ω_i and δ_i are constants. Of course, (7) is trivially satisfied, whereas (8) determines the *dispersion law*,

$$\omega_i = k_i^3 \ , \qquad i = 1,2,3. \tag{14}$$

The terms generated by $B(f_1, f_1)$ in (9) justify the choice

$$\begin{aligned} f_2 &= a_{12}\exp(\theta_1+\theta_2) + a_{13}\exp(\theta_1+\theta_3) + a_{23}\,exp(\theta_2+\theta_3) \\ &= a_{12}\exp\left[(k_1+k_2)x - (\omega_1+\omega_2)t + \delta_1+\delta_2\right] \\ &+ a_{13}\exp\left[(k_1+k_3)x - (\omega_1+\omega_3)t + \delta_1+\delta_3\right] \\ &+ a_{23}\exp\left[(k_2+k_3)x - (\omega_2+\omega_3)t + \delta_2+\delta_3\right] \ , \end{aligned} \tag{15}$$

and (9) allows to calculate the constants a_{12}, a_{13} and a_{23}. With (14) one obtains

$$a_{ij} = \frac{(k_i - k_j)^2}{(k_i + k_j)^2} \ , \qquad i,j = 1,2,3. \tag{16}$$

Then, $B(f_1{\cdot}f_2 + f_2{\cdot}f_1)$ in (10) motivates the particular solution

$$\begin{aligned} f_3 &= b_{123}\exp(\theta_1+\theta_2+\theta_3) \\ &= b_{123}\exp\left[(k_1+k_2+k_3)x - (\omega_1+\omega_2+\omega_3)t + \delta_1+\delta_2+\delta_3\right], \end{aligned} \tag{17}$$

and one calculates

$$b_{123} = a_{12}\, a_{13}\, a_{23} = \frac{(k_1-k_2)^2 (k_1-k_3)^2 (k_2-k_3)^2}{(k_1+k_2)^2 (k_1+k_3)^2 (k_2+k_3)^2} \ . \tag{18}$$

Subsequently, (11) allows to verify that indeed $f_4 = 0$. In the sixth equation of the scheme $B(f_2{\cdot}f_3 + f_3{\cdot}f_2)$ should equal zero in order to assure that $f_5 = 0$. If so, it will be possible to take $f_i = 0$ for $i > 6$. Finally setting $\epsilon = 1$ in (6), we obtain

$$\begin{aligned} f &= 1 + \exp\theta_1 + \exp\theta_2 + \exp\theta_3 \\ &+ a_{12}\exp(\theta_1+\theta_2) + a_{13}\exp(\theta_1+\theta_3) + a_{23}\exp(\theta_2+\theta_3) \\ &+ b_{123}\exp(\theta_1+\theta_2+\theta_3) \ , \end{aligned} \tag{19}$$

which upon substitution in (2) generates the well-known three soliton solution of (1).

The construction of N-soliton solutions [1-6] with $N \geq 3$ is tedious and the necessary algebraic simplifications and factorizations are bound to fail if carried out by hand. Hence the need for a symbolic program that relieves us of the elaborate calculations.

III. MACSYMA PROGRAM FOR THE HIROTA METHOD

This preliminary program calculates the one, two and three soliton solutions of a fairly simple completely integrable PDE, provided it can be transformed into a *single* bilinear equation for the new variable f. The program is written in such a way that the extension for the N-soliton is straightforward. The structure should also allow to 'translate' it into the language of e.g. MATHEMATICA, MAPLE or REDUCE. The user must select the value of N and also provide the bilinear operator B for the PDE.

```
    /* MACSYMA program for the HIROTA METHOD */
writefile("three_soliton_sawada_kotera.out");
n:3;
B(f,g):=Dxt[1,1](f,g)+Dx[6](f,g)$
showtime:true $
depends([f,g],[x,y,t])$
Dx[n](f,g):=sum((-1)^(n-j)*n!*diff(f,x,j)*diff(g,x,n-j)
/(j!*(n-j)!),j,0,n)$
Dy[n](f,g):=sum((-1)^(n-j)*n!*diff(f,y,j)*diff(g,y,n-j)
/(j!*(n-j)!),j,0,n)$
Dxt[m,n](f,g):=sum((-1)^(m-j)*m!*sum((-1)^(n-i)*n!*
diff(diff(f,x,j),t,i)*diff(diff(g,x,m-j),t,n-i)
/(i!*(n-i)!),i,0,n)/(j!*(m-j)!),j,0,m)$
for i:1 thru n do f[i]:0 $
f[1]:sum(exp(th(i,x,y,t)),i,1,n)$
gradef(th(i,x,y,t),0,k[i],l[i],-om[i])$
bonf1:expand(ev(B(1,f[1]),diff))$
bonf1:expand(ev(bonf1))$
for i:1 thru n do (eqone[i]:ratcoef(bonf1,exp(th(i,x,y,t)),1),
om[i]:factor(rhs(part(solve(eqone[i],om[i]),1))),
if n>1 then(tf[2]:sum(a[i,j]*exp(th(i,x,y,t))*exp(th(j,x,y,t)),j,i+1,n),
f[2]:ev(f[2]+tf[2])))$
om[1];
om[2];
om[3];
bonf2:expand(ev(b(1,f[2])+b(f[1],f[1])+b(f[2],1),diff))$
bonf2:expand(ev(bonf2))$
if n>1 then(for i:1 thru n do (for j:i+1 thru n do(
eqtwo[i,j]:ratcoef(bonf2,exp(th(i,x,y,t))*exp(th(j,x,y,t)),1),
a[i,j]:factor(rhs(part(solve(eqtwo[i,j],a[i,j]),1))),
if n>2 then(tf[3]:sum(b[i,j,k]*exp(th(i,x,y,t))*exp(th(j,x,y,t))*
exp(th(k,x,y,t)),k,j+1,n),
f[3]:ev(f[3]+tf[3])))))$
a[1,2];
a[1,3];
a[2,3];
bonf3:expand(ev(b(1,f[3])+b(f[1],f[2])+b(f[2],f[1])+b(f[3],1),diff))$
bonf3:expand(ev(bonf3))$
length(bonf3);
if n>2 then(for i:1 thru n do (for j:i+1 thru n do (for k:j+1 thru n
do(eqthree[i,j,k]:ratcoef(bonf3,exp(th(i,x,y,t))*exp(th(j,x,y,t))*
exp(th(k,x,y,t)),1),
b[i,j,k]:factor(rhs(part(solve(eqthree[i,j,k],b[i,j,k]),1)))))))$
f:1+sum(f[i],i,1,n);
b[1,2,3];
closefile();
quit();
```

IV. EXAMPLES AND TEST CASES

- For the KdV equation [1-6], $u_t + 6uu_x + u_{3x} = 0$, one uses (2) and $B(f,g) := Dxt[1,1](f,g) + Dx[4](f,g)$.
  The output of the program confirmed the results in (14), (16) and (18).

- For the Kadomtsev-Petviashvili equation [2-6], $(u_t + 6uu_x + u_{3x})_x + 3u_{2y} = 0$, one has $u = 2\partial_x^2 \ln f(x,y,t)$ and $B(f,g) := Dxt[1,1](f,g) + Dx[4](f,g) + 3*Dy[2](f,g)$.
  In this case $\theta_i = k_i x + l_i y - \omega_i t$, and for simplicity we selected $l_i = k_i m_i \, (i = 1,2,3)$.
  Note: insert the line $l[i] := k[i] * m[i]$ at the beginning of the program. We obtain $\omega_i = k_i(k_i^2 + 3m_i^2)$, $i = 1,2,3$, and
  $$a_{ij} = \frac{(k_i - k_j - m_i + m_j)(k_i - k_j + m_i - m_j)}{(k_i + k_j + m_i - m_j)(k_i + k_j - m_i + m_j)}, \quad i,j = 1,2,3. \tag{20}$$
  The program could not calculate $b_{123} = a_{12}a_{13}a_{23}$ in a reasonable amount of time because the equation for b_{123} has 267 terms in $\exp(\theta_1 + \theta_2 + \theta_3)$.

- For the Boussinesq equation [1-6], $u_{2t} - u_{2x} - 3(u^2)_{2x} - u_{4x} = 0$, one has again (2) and
  $B(f,g) := Dxt[0,2](f,g) - Dx[2](f,g) - Dx[4](f,g)$.
  The results then are $\omega_i = -k_i\sqrt{1 + k_i^2}$, $i = 1,2,3$, and
  $$a_{ij} = \frac{\sqrt{1+k_i^2}\sqrt{1+k_j^2} - 2k_i^2 + 3k_ik_j - 2k_j^2 - 1}{\sqrt{1+k_i^2}\sqrt{1+k_j^2} - 2k_i^2 - 3k_ik_j - 2k_j^2 - 1}, \quad i,j = 1,2,3. \tag{21}$$
  The program did also determine the explicit form of $b_{123} = a_{12}a_{13}a_{23}$.

- For the Sawada-Kotera equation [2,5,6], $u_t + 45u^2u_x + 15u_xu_{2x} + 15uu_{3x} + u_{5x} = 0$, one has (2) and $B(f,g) := Dxt[1,1](f,g) + Dx[6](f,g)$.
  Furthermore, $\omega_i = k_i^5$, $i = 1,2,3$. The coefficients are given by
  $$\begin{aligned} a_{ij} &= \frac{(k_i - k_j)^2(k_i^2 - k_ik_j + k_j^2)}{(k_i + k_j)^2(k_i^2 + k_ik_j + k_j^2)} \\ &= \frac{(k_i - k_j)^3(k_i^3 + k_j^3)}{(k_i + k_j)^3(k_i^3 - k_j^3)}, \qquad i,j = 1,2,3, \quad (22) \\ b_{123} &= a_{12}a_{13}a_{23}. \quad (23) \end{aligned}$$

- For the shallow water wave equation [5], $u_{xxt} + 3uu_t - 3u_x \int_x u_t \, dx' - u_x - u_t = 0$, one uses $u = \partial_x^2 \ln f$ and $B(f,g) := Dxt[3,1](f,g) - Dx[2](f,g) - Dxt[1,1](f,g)$.
  The program calculated that
  $$\begin{aligned} \omega_i &= \frac{k_i}{(1+k_i)(1-k_i)}, \qquad i = 1,2,3, \quad (24) \\ a_{ij} &= \frac{(k_i - k_j)^2(k_i^2 - k_ik_j + k_j^2 - 3)}{(k_i + k_j)^2(k_i^2 + k_ik_j + k_j^2 - 3)}, \; i,j = 1,2,3. \quad (25) \end{aligned}$$
  The program could not determine $b_{123} = a_{12}a_{13}a_{23}$ due to the large number of terms in f_3.

Hirota's bilinear operators used in these examples, $Dxt[m,n](f,g)$, $Dx[n](f,g)$, and $Dy[n](f,g)$ are defined in the program itself.

REFERENCES

[1] R. Hirota, in: *Bäcklund Transformations, the Inverse Scattering Method, Solitons, and Their Applications*, Lecture Notes in Mathematics 515, ed. R.M. Miura, Springer-Verlag, Berlin, 1976, pp. 40-68.

[2] R. Hirota, in: *Solitons*, Topics in Physics 17, eds. R.K. Bullough and P.J. Caudrey, Springer-Verlag, Berlin, 1980, pp. 157-76.

[3] M.J. Ablowitz and H. Segur, *Solitons and the Inverse Scattering*, SIAM Studies in Applied Mathematics 4, SIAM, Philadelphia, 1981.

[4] P.G. Drazin and R.S. Johnson, *Solitons: an introduction*, Cambridge University Press, Cambridge, 1989.

[5] J. Hietarinta, *A search for bilinear equations passing Hirota's three-soliton condition*, Parts I-IV, J. Math. Phys. 28, 1732-42, 1987, *ibid.* 2094-101, 1987, *ibid.* 2586-92, 1987, *ibid.* 29 628-35, 1988.

[6] J. Hietarinta, in : *Partially Integrable Evolution Equations in Physics*, Proceedings of the Summer School for Theoretical Physics, Les Houches, France, March 21-28, 1989, eds: R. Conte and N. Boccara, Kluwer Academic Publishers, pp. 459-78, 1990.

A DIFFERENTIAL-DIFFERENCE EQUATION FOR HIGHER NONLINEAR SCHRÖDINGER EQUATION

Thiab R. Taha
Computer Science Department
University of Georgia
415 GSRC
Athens, GA 30602 U.S.A.

Abstract - A new differential difference equation is obtained which has as its limiting form a higher nonlinear Schrödinger (HNLS) equation. This new equation is constructed by methods related to the inverse scattering transform (IST) and can be used as a numerical scheme for the HNLS equation.

1. INTRODUCTION

In 1975 Ablowitz and Ladik proposed a new discrete eigenvalue problem, an appropriate generalization of a discretized version of the eigenvalue problem of Zakharov and Shabat, as a basis for generating solvable discrete equations [1]. They derived discrete (differential-difference) versions of the nonlinear Schrödinger (NLS), Korteweg-de Vries (KdV), modified KdV, and "sine-Gordon" equations.

In 1984 Taha and Ablowitz derived differential-difference and partial difference equations for KdV, and MKdV equations [2]. In this paper a differential-difference equation is obtained which has as its limiting fom the HNLS equation

$$iq_t = q_{xxx} + 8q_{xx}|q|^2 + 4qq_xq_x^* + 6q^*(q_x)^2$$

$$+ 2q^2q_{xx}^* + 6|q|^4q \qquad (1)$$

This differential-difference equation can be used as a numerical scheme for the HNLS equation (1).

II. DERIVING NONLINEAR DIFFERENTIAL-DIFFERENCE EQUATION

The key step in obtaining differential-difference equations which can be solved by the inverse scattering transform is to make an association between the nonlinear evolution equation and a linear eigenvalue (scattering) problem. To find a nonlinear differential-difference equation associated with the HNLS equation, it is essential to use (a) a suitable eigenvalue problem e.g.,

$$V_{1n+1} = z\,V_{1n} + Q_n(t)\,V_{2n}$$

$$V_{2n+1} = \frac{1}{z}\,V_{2n} + R_n(t)\,V_{1n} \qquad (2)$$

where z is the eigenvalue and the potentials R_n, Q_n are defined on the spacelike interval $|n| < \infty$ and the time $t > 0$, and (b) the associated time dependence,

$$V_{1nt} = A_nV_{1n} + B_n\,V_{2n}\;,$$

$$V_{2nt} = C_n\,V_{1n} + D_n\,V_{2n} \qquad (3)$$

where the functions A_n, B_n, C_n, D_n depending in general on the potentials. The equations for determining the sets A_n, B_n, C_n, D_n and hence the evolution equation are obtained by requiring the eigenvalue z to be time invariant ($\frac{\partial z}{\partial t} = 0$) and by forcing the consistency

$$\frac{\partial}{\partial t}(E\,V_{in}) = E(V_{int}),\; i = 1,2 \qquad (4)$$

where E is the shift operator defined by $E\,V_{in} = V_{in+1}$, $i = 1, 2$. Performing the operations indicated in (4) results in four equations which are given by

$$z(\Delta_nA_n) - C_nQ_n + R_nB_{n+1} = 0$$

$$\frac{1}{z}\,B_{n+1} - zB_n + Q_n(A_{n+1} - D_n) = Q_{nt}$$

$$z\,C_{n+1} - R_nA_n + R_nD_{n+1} - \frac{1}{z}c_n = R_{nt} \qquad (5)$$

$$\frac{1}{z}(\Delta_nD_n) + C_{n+1}\,Q_n - R_n\,B_n = 0$$

where $\Delta_n\,A_n = A_{n+1} - A_n$, etc.

Using the ideas in [1,2,3], the coefficients for the time dependence of the eigenfunctions are expanded as follows:

$$A_n = \sum_{k=-2}^{2} z^{2k}\,A_n^{(2k)},\; B_n = \sum_{k=-1}^{2} z^{(2k-1)}\,B_n^{(2k-1)},$$

$$C_n = \sum_{k=-1}^{2} z^{(2k-1)}\,C_n^{(2k-1)},\; D_n = \sum_{k=-2}^{2} z^{2k}\,D_n^{(2k)}. \qquad (6)$$

With the expanded form of A_n, B_n, C_n, D_n, (5) yields a system of twenty equations in eighteen unknowns corresponding to equating powers of $z^5, z^{-5}, z^4, z^{-4}, \cdots, z, z^{-1}$, all of which must be independently satisfied. Carrying out the algebra we find the values of $A_n^{(4)}$, , $D_n^{(-4)}$ in terms of the potentials [2]. The remaining two equations are the evolution equations. For the special case associated with the NLS and HNLS equation we let $R_n = -Q_n^*$ (where Q_n^* is the complex conjugate of Q_n) then the remaining two coupled equations are consistent under the conditions:

$$A_-^{(4)} - D_-^{(4)} = D_-^{(-4)*} - A_-^{(-4)*} = \alpha\;,$$

$$A_-^{(2)} - D_-^{(2)} = D_-^{(-2)*} - A_-^{(-2)*} = \beta\;, \qquad (7)$$

$$A_-^{(0)} - D_-^{(0)} = D_-^{(0)*} - A_-^{(0)*} = \gamma$$

with $\alpha = -\alpha^*, \beta = -\beta^*$, and $\gamma = -\gamma^*$ and the nonlinear differential-difference equation is

$$Q_{nt} = (1 + |Q_n|^2)\Big[\beta(Q_{n+1} + Q_{n-1}) + \alpha\Big\{(Q_{n+2} + Q_{n-2})$$

$$+ Q_{n+2}\,|Q_{n+1}|^2 + Q_{n-2}\,|Q_{n-1}|^2 + Q_n^*(Q_{n+1}^2 + Q_{n-1}^2)$$

$$+ Q_n(Q_{n+1}\,Q_{n-1}^* + Q_{n-1}Q_{n+1}^*)\Big\}\Big] + \gamma\,Q_n \qquad (8)$$

Let $Q_n = \Delta x\, q_n$, taking the limit as $\Delta x \rightarrow 0$ in (8) and by a proper choice of the constants

(a) $\alpha = \frac{i}{12(\Delta x)^2}$, $\beta = \frac{-16i}{12(\Delta x)^2}$, and $\gamma = \frac{30i}{12(\Delta x)^2}$ yields the NLS equation

$$iq_t = q_{xx} + 2|q|^2 q \; ; \tag{9}$$

(b) $\alpha = \frac{-i}{(\Delta x)^4}$, $\beta = \frac{4i}{(\Delta x)^4}$, and $\gamma = \frac{-6i}{(\Delta x)^4}$ yields the HNLS equation (1).

III. CONCLUSION

The differential-difference equations derived in this paper have as limiting forms the NLS and HNLS equations. These equations can be used as numerical schemes for the associated nonlinear evolution equations. These IST schemes have a number of special properties [2].

IV. ACKNOWLEDGEMENTS

This research has been supported in part by the Department of Energy Grant No. DE-FG09-90ER25088.

REFERENCES

1. Ablowitz, M. J., and Ladik, J. F., 'Nonlinear differential-difference equations', Jour. Math. Phys. Vol. 16, #3, (1975), 593-603.

2. Taha, T. R., and Ablowitz, M. J., 'Analytical and Numerical Aspects of Certain Nonlinear Evolution Equations. I. Analytical', J. Comp. Phys. Vol. 55, No. 2 (1984), 192-202.

3. Ablowitz, M. and Segur, H., Solitons and the inverse scattering transform, SIAM, Philadelphia, 1981.

Nonlinear Evolution Equations and Painlevé Test

W.-H. Steeb and N. Euler

Department of Applied Mathematics and Nonlinear Studies
Rand Afrikaans University, Johannesburg 2000, South Africa

Abstract: The Painlevé test is a powerful tool [1-6] in the study of nonlinear evolution equations. We can study the integrability of ordinary and partial differential equations. Exact solutions can be constructed. Bäcklund transformations [illegible] can be derived within this approach. By making use of the Painlevé test the construction of Lie Bäcklund symmetries is straightforward. It also plays a fundamental role in the investigation of the chaotic behaviour for ordinary differential equations. We apply the Painlevé test to the anharmonic oscillator, the semi-classical Jaynes-Cumming model, the energy eigenvalue level motion equation, the Katomdsev-Petviashvili equation, the nonlinear Klein-Gordon equation and the self-dual Yang-Mills equation and its connection with the Yang-Mills equation.

First we discuss ordinary differential equations. The ordinary differential equations are extended into the complex domain.

Definition: An ordinary differential equation is said to have the Painlevé property when every solution is single valued, except at the fixed singularities of the coefficients. That is, the Painlevé property requires that the movable singularities are no worse than poles.

Theorem: A necessary condition that an n-th order ordinary differential equation of the form $dw/dz = g(w)$ where g is rational in w has the Painlevé property is that there is a Laurent expansion

$$w_k(z) = (z - z_1)^{m_k} \sum_{j=0}^{\infty} a_{kj}(z - z_1)^j$$

with $n - 1$ arbitrary expansion coefficients (besides the pole position which is arbitrary).

In the following we give two examples where we apply the Painlevé test to find solutions.

Example 1. The semiclassical Jaynes-Cummings model is given by

$$\begin{aligned} \frac{dS_1}{dt} &= -S_2, \quad \frac{dS_2}{dt} = S_1 + S_3 E \\ \frac{dS_3}{dt} &= -S_2 E, \quad \frac{d^2 E}{dt^2} + \mu^2 E = \alpha S_1 \end{aligned}$$

where μ and α are constants. There is numerical evidence that the system shows chaotic behaviour for certain parameter values and initial conditions. To perform the Painlevé test we have to consider the system in the complex domain. For the sake of simplicity we do not change our notation. First we look for the dominant behaviour. Inserting the ansatz $(j = 1, 2, 3)$

$$S_j(t) \propto S_{j,0}(t - t_1)^{m_j}, \quad E(t) \propto E_0(t - t_1)^{m_4}$$

into the Jaynes-Cummings model we find that the sytem with the dominant terms is given by

$$\frac{dS_1}{dt} = -S_2, \quad \frac{dS_2}{dt} = S_3 E, \quad \frac{dS_3}{dt} = -S_2 E, \quad \frac{d^2 E}{dt^2} = \alpha S_1$$

and $m_1 = -3$, $m_2 = -4$, $m_3 = -4$ and $m_4 = -1$. For the expansion coeffients we obtain $S_{1,0} = \frac{8i}{\alpha}$, $S_{2,0} = \frac{24i}{\alpha}$, $S_{3,0} = -\frac{24}{\alpha}$, $E_0 = 4i$. From the dominant behaviour we conclude that the system with the dominant terms is scale invariant under $t \to \epsilon^{-1} t$, $S_1 \to \epsilon^3 S_1$, $S_2 \to \epsilon^4 S_2$, $S_3 \to \epsilon^4 S_3$, $E \to \epsilon E$. Next we determine the resonances and the Kowalevski exponents. Inserting the ansatz $(j = 1, 2, 3)$

$$\begin{aligned} S_j(t) &= S_{1,0}(t - t_1)^{m_j} + A(t - t_1)^{m_j + r} \\ E(t) &= E_0(t - t_1)^{-1} + D(t - t_1)^{-1+r} \end{aligned}$$

into the system with the dominant terms we find the resonances $-1, 4, 8, 3/2 \pm i\sqrt{15}/2$. The Kowalevski exponents can be found from the variational equation. We find that the resonances and the Kowalevski exponents coincide. The two Kowalevski exponents 4 and 8 can be related to first integrals. We obtain $I_1 = S_2^2 + S_3^2$ and $I_2 = \alpha S_3 - \alpha S_1 E + (dE/dt)^2/2$ since $I_1(\epsilon^3 S_1, \epsilon^4 S_2, \epsilon^4 S_3, \epsilon^1 E) = \epsilon^8(S_2^2 + S_3^2)$, $I_2(\epsilon^3 S_1, \epsilon^4 S_2, \epsilon^4 S_3, \epsilon^1 E) = \epsilon^4(\alpha S_3 - \alpha S_1 E + (dE/dt)^2/2)$. Using the first integrals for the system with the dominant terms we find on inspection that the first integrals for the Jaynes-Cummings model are given by $I_1 = S_1^2 + S_2^2 + S_3^2$ and $I_2 = \alpha S_3 - \alpha S_1 E + 1/2\mu^2 E^2 + (dE/dt)^2/2$. From the Painlevé test we find that the Jaynes-Cummings model admits a Laurent expansion of the form $(j = 1, 2, 3)$

$$S_j(t) = \sum_{i=0}^{\infty} S_{j,i}(t - t_1)^{i+m_j}, \quad E(t) = \sum_{i=0}^{\infty} E_i(t - t_1)^{i-1}$$

with three arbitrary constants (including t_1). The expansion coefficients are determined by a recursion relation. In particular we find $S_{1,1} = S_{2,1} = S_{3,1} = E_1 = 0$. This local expansion is not the general solution (owing to the complex resonances) which requires five arbitrary constants. We now construct an exact solution from the Laurent expansions (9) through (12). Let k be the modulus of the elliptic functions $\text{sn}(z,k)$, $\text{cn}(z,k)$ and $\text{dn}(z,k)$. We define $K'(k) := \int_0^1 (1 - t^2)^{-1/2}(1 - k'^2 t^2)^{-1/2} dt$ where $k'^2 := 1 - k^2$. By the addition-theorem of the Jacobi elliptic functions, we have $\text{sn}(z + iK', k) = 1/k\text{sn}(z,k)$. Similarly $\text{cn}(z + iK', k) = -i/k\text{dn}(z,k)/\text{sn}(z,k)$, $\text{dn}(z + iK', k) = -i\text{cn}(z,k)/\text{sn}(z,k)$. For points in the neighbourhood of the point $z = 0$, the function $\text{sn}(z,k)$ can be expanded by Taylor's theorem in the form $\text{sn}(z,k) = \text{sn}(0,k) + z\text{sn}'(0,k) + 1/2z^2\text{sn}''(0,k) + 1/3!z^3\text{sn}'''(0,k) + \cdots$ where accents denote derivatives. ce $\text{sn}(0,k) = 0$, $\text{sn}'(0,k) = 1$, $\text{sn}''(0,k) = 0$, $\text{sn}'''(0,k) = -(1 + k^2)$ etc. the expansion becomes $\text{sn}(z,k) = z - \frac{1}{6}(1 + k^2)z^3 + \cdot$. Therefore $\text{cn}(z,k) = (1 - \text{sn}^2 z)^{1/2} = 1 - 1/2z^2 + \cdots$ and $\text{dn}(z,k) = (1 - k^2\text{sn}^2 z)^{1/2} = 1 - 1/2k^2 z^2 + \cdots$. Consequently

$$\text{sn}(z + iK', k) = \frac{1}{k\text{sn}(z,k)} == \frac{1}{kz} + \frac{1 + k^2}{6k} z + \frac{1}{6^2 k}(1 + k^2)^2 z^3 + \cdots .$$

Similarly, we find that $\text{cn}(z + iK', k) = \frac{-i}{kz} + \frac{2k^2 - 1}{6k} iz + \cdots$ and $\text{dn}(z + iK', k) = -\frac{i}{z} + \frac{2 - k^2}{6} iz + \cdots$. It follows that at the point $z = iK'$; the functions $\text{sn}(z,k)$, $\text{cn}(z,k)$ and $\text{dn}(z,k)$ have simple poles, with the residues $1/k$, $-i/k$, $-i$, respectively. We can focus our attention to the quantity E since S_1 can be derived from E. Then the quantities S_2 and S_3 can be found. Comparing the Laurent expansion and the expansion of the elliptic functions we find that the Jaynes-Cummings model admits the particular solution (in the real domain) $E(t) = E_0 \mathrm{dn}(\Omega t, k)$ where $E_0^2 = 16\Omega^2$, $k = 2(1+(1-\sqrt{1+c})/c)$, $\Omega^2 = c(\mu^2 - \frac{1}{3})/(4(\sqrt{1+c}-1))$ and

$$c = -(\mu^2 - \frac{1}{3})^{-2}\left\{\frac{4}{3}\left[\alpha^2 - 4(\mu^2 - \frac{1}{9})^3\right]^{1/2} + (\mu^2 - \frac{1}{9})(\mu^2 - \frac{17}{9})\right\}.$$

The quantities S_1, S_2 and S_3 can now easily be found from Jaynes-Cummings model.

Example 2: In the study of energy level motion of the Hamilton operator we arrive for a two level system at the following system of ordinary differential equations

$$\frac{dE}{d\lambda} = p, \quad \frac{dp}{d\lambda} = 4\frac{V^2}{E}, \quad \frac{dV}{d\lambda} = -\frac{Vp}{E}.$$

To perform the Painlevé test we consider the system in the complex domain. Inserting the ansatz $E(\lambda) \propto E^{(0)}(\lambda-\lambda_0)^m$, $p(\lambda) \propto p^{(0)}(\lambda-\lambda_0)^n$, $V(\lambda) \propto V^{(0)}(\lambda-\lambda_0)^q$ and comparing the exponents yields $n = q = m-1$, where m is arbitrary at this stage. Obviously, all terms are dominant. Therefore, the dynamical system is scale invariant under $\lambda \to \varepsilon^{-1}\lambda$, $\quad E \to \varepsilon^{-m}E$, $p \to \varepsilon^{-m+1}p$, $V \to \varepsilon^{-m+1}V$. Next we determine the coefficients $E^{(0)}$, $p^{(0)}$, and $V^{(0)}$. Requiring that $E^{(0)}$, $p^{(0)}$, $V^{(0)} \neq 0$, we find $m = 1/2$, $n = q = -1/2$. Furthermore, $p^{(0)} = E^{(0)}/2$ and $V^{(0)} = \pm iE^{(0)}/4$ with $E^{(0)}$ arbitrary. Consequently, $E(\lambda) = E^{(0)}(\lambda - \lambda_0)^{1/2}$, $p(\lambda) = E^{(0)}/2(\lambda-\lambda_0)^{1/2}$, $V(\lambda) = \mp iE^{(0)}/4(\lambda-\lambda_0)^{-1/2}$ is a solution, where $E^{(0)}$ and λ_0 are arbitrary. However, it is not the general solution which requires three arbitrary constants. When we determine the resonances, using this solution, we obtain $-1, 0, 1$. When we determine the Kowalewski exponents, using this solution, we also find $-1, 0, 1$. The Kowalewski exponents $0, 1$ can be associated with the (polynomial) first integrals of the dynamical system. On inspection, we find that $I_1(E,p,V) = p^2/4 + V^2$, $I_2(E,p,V) = EV$ are first integrals. Then we obtain the scaling behaviour

$$\begin{aligned} I_1(\varepsilon^{-1/2}E, \varepsilon^{1/2}p, \varepsilon^{1/2}V) &= \varepsilon^1 I_1(E,p,V) \\ I_2(\varepsilon^{-1/2}E, \varepsilon^{1/2}p, \varepsilon^{1/2}V) &= \varepsilon^0 I_2(E,p,V) \end{aligned}$$

where the exponents of ε give the Kowalewski exponents, namely 0 and 1. Thus, the dynamical system is algebraic completely integrable. Inserting the expansion

$$\begin{aligned} E(\lambda) &= (\lambda-\lambda_0)^{1/2}\sum_{j=0}^{\infty} E^{(j)}(\lambda-\lambda_0)^j \\ p(\lambda) &= (\lambda-\lambda_0)^{-1/2}\sum_{j=0}^{\infty} p^{(j)}(\lambda-\lambda_0)^j \\ V(\lambda) &= (\lambda-\lambda_0)^{-1/2}\sum_{j=0}^{\infty} V^{(j)}(\lambda-\lambda_0)^j \end{aligned}$$

into the given dynamical system, we find one more arbitrary constant (besides $E^{(0)}$ and λ_0), so that the expansion represents the general solution. The system does not pass the Painlevé test, due to the dominant behavior. However, it passes the so-called quasi-Painlevé test, i.e., it admits an expansion of the form given above with three arbitrary constants Obviously, we can also find the general real solution to the dynamical system, namely

$$E^2(\lambda) = \frac{I_2^2}{I_1} + 4I_1(\lambda-\lambda_0)^2, \quad p^2(\lambda) = \frac{16I_1^3(\lambda-\lambda_0)^2}{I_2^2 + 4I_1^2(\lambda-\lambda_0)^2}$$

$$V^2(\lambda) = \frac{I_1 I_2^2}{I_2^2 + 4I_1^2(\lambda-\lambda_0)^2}$$

The general solution contains three free parameters I_1, I_2, and λ_0 which are determined from the initial values $E(\lambda = 0)$, $p(\lambda = 0)$, and $V(\lambda = 0)$.

Next we consider partial differential equations. Suppose that there are n independent variables, and that the system of partial differential equations has coefficients that are holomorphic on C^n. We cannot simply require that all the solutions of this system be meromorphic on C^n, since arbitrarily nasty singularities can occur along characteristic hypersurfaces. The following definition [2] of the Painlevé property avoids this problem. *If S is a holomorphic non-characteristic hypersurface in C^n, then every solution that is holomorphic on $C^n \backslash S$ extends to a meromorphic solution on C^n.*

In other words, if a solution has a singularity on a non characteristic hypersurface, then that singularity is a pole and nothing worse. A slightly weaker form of the Painlevé property was formulated by Weiss et al [1]. It involves looking for solutions ϕ of the system of partial differential equations in the form

$$u = \phi^{-\alpha}\sum_{n=0}^{\infty} u_n \phi^n$$

where ϕ is a holomorphic function whose vanishing defines a non-characteristic hypersurface. Substituting this series into the partial differential equations yields conditions on the number α and recursion relations for the functions u_n. The requirement is that α should turn out to be a non-negative integer, and the recursion relations should be consistent, and that the series expansion should contain the correct number of arbitrary functions (counting ϕ as one of them).

It has been found that integrable equations satisfy this weaker form (perhaps after a change of variables), whereas non-integrable equations fail it. To establish Painlevé property is more difficult (the Painlevé property implies the weaker form, but the reverse implication need not hold in general). However, it seems that in practice, the weaker form is sufficient to ensure integrability.

The Painlevé property seems to be a useful indicator of integrability or solvability, or both. Integrable here means ther exists a nontrivial Lax Pair for the system.

Example 1: The Kadomtsev-Petviashvili equation is given by

$$\frac{\partial^2 u}{\partial t\partial x} + \left(\frac{\partial u}{\partial x}\right)^2 + u\frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} + 3\sigma^2\frac{\partial^2 u}{\partial y^2} = 0$$

where $\sigma^2 = \pm 1$. The Kadomtsev Petviashvili equation is a completely integrable soliton equation. The Katomdsev Petvialshilvi equation has the Painlevé property. We consider the generalized variable-coefficient Kadomtsev Petviashvili equation

$$\begin{aligned} \frac{\partial^2 u}{\partial t\partial x} + \left(\frac{\partial u}{\partial x}\right)^2 + u\frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} + a(y,t)\frac{\partial u}{\partial x} + b(y,t)\frac{\partial u}{\partial y} \\ + c(y,t)\frac{\partial u}{\partial y} + d(y,t)\frac{\partial^2 u}{\partial x\partial y} + e(y,t)\frac{\partial^2 u}{\partial x^2} &= 0 \end{aligned}$$

where $a(y,t)$, $b(y,t)$, $c(y,t)$, $d(y,t)$, and $e(y,t)$ are analytic functions. In order that the generalized Kadomtsev Petviashvili equation satisfy the Painlevé property the function a, b, c, d, ϵ have to satisfy the system of equations

$$\begin{aligned}\frac{\partial a}{\partial t}+2a^2+d\frac{\partial a}{\partial y}-\frac{1}{2}\frac{\partial c}{\partial y}\frac{\partial e}{\partial y}-c\frac{\partial^2 e}{\partial y^2} &= 0\\ \frac{\partial c}{\partial t}+4ac-2c\frac{\partial d}{\partial y}+d\frac{\partial c}{\partial y} &= 0\\ (2b-\frac{\partial c}{\partial y})c=0, \quad b(2b-\frac{\partial c}{\partial y}) &= 0.\end{aligned}$$

These equations are necessary and sufficient conditions for the generalized Kadomtsev Petviashvili equations to be transformable into the Kadomtsev Petviashvili equation, provided that $c \neq 0$. If $c = 0$, the equation may be transformed into the Korteweg de Vries equation provided that $b = 0$ and $\partial a/\partial t + d\partial a/\partial y + 2a^2 = 0$. *Example 2.* Consider the nonlinear Klein Gordon equation $\partial^2 v/\partial\xi\partial\eta = v^3$. Inserting the expansion

$$v = \phi^m \sum_{j=0}^{\infty} v_j \phi^j$$

we find $m = -1$, $v_0^2 = 2\phi_\xi\phi_\eta$ and

$$\begin{aligned}-\phi_{\eta\xi}v_0-\phi_\eta v_{0\xi}-\phi_\xi v_{0\eta} &= 3v_0^2v_1\\ v_{0\eta\xi} &= 3v_0^2v_2+3v_0v_1^2\\ 2\phi_\eta\phi_\xi v_3+\phi_{\xi\eta}v_2+\phi_\eta v_{2\xi}+\phi_\xi v_{2\eta}+v_{1\eta\xi} &= 2v_0^2v_3+6v_0v_1v_2.\end{aligned}$$

At the resonance $r = 4$ we obtain

$$2\phi_{\eta\xi}v_3+2\phi_\eta v_{3\xi}+2\phi_\xi v_{3\eta}+v_{2\eta\xi} = 6v_0v_1v_3+3v_2v_1^2+3v_0v_2^2.$$

Inserting the equations for the coefficients into the above equation for the resonance we obtain a "huge" partial differential equation for ϕ [6]. If ϕ satisfies this condition, then the expansion coefficient $v_4(\xi,\eta)$ is arbitrary. The nonlinear Klein Gordon equation admits the symmetry vector fields $\{\partial/\partial\xi, \partial/\partial\eta, \partial/\partial\xi - \partial/\partial\eta, \partial/\partial\xi + \partial/\partial\eta - v\partial/\partial v\}$.

Next we construct similarity ansätze:

(1) From $\partial/\partial\eta$ and $\partial/\partial\xi$ we find $s = c_1\xi + c_2\eta$ and $v(\eta,\xi) = f(s)$. We obtain $d^2f/ds^2 = f^3/(c_1c_2)$. This equation passes the Painlevé test. Therefore $\phi(\eta,\xi) = c_1\eta + c_2\xi$ satisfy the condition on ϕ.

(2) The symmetry generator $\xi\partial/\partial\xi - \eta\partial/\partial\eta$ leads to the similarity ansatz $v(\eta,\xi) = f(s)$ with $s = \eta\xi$. It follows that $d^2f/ds^2 + (df/ds)/s - f^3/s = 0$. This equation does not pass the Painlevé test. This is in agreement that $\phi(\xi,\eta) = \xi\eta$ does not satisfy the conditional equation on ϕ.

(3) The symmetry generator $\xi\partial/\partial\xi + \eta\partial/\partial\eta - v\partial/\partial v$ leads to the similarity ansatz $v(\eta,\xi) = f(s)/\xi$ with $s = \eta/\xi$. It follows that $d^2f/ds^2 + 2(dfds)/s + f^3/s = 0$. This equation passes the Painlevé test. This is in agreement that $\phi(\eta,\xi) = \eta/\xi$ satisfy the conditional equation on ϕ.

Exanple 3: We now consider the self-duality Yang Mills equations. Let G be a Lie group (the 'gauge group') and g it Lie algebra. A gauge potential (connection) A is a g-valued 1-form on $\mathcal{R}^4$. The corresponding gauge field (curvature) is the g-valued 2-form $F := DA \equiv dA + [A,A]$ where DA is the covariant exterior derivative of A. Two gauge potentials A and A' are regarded as being equivalent if they are related by a gauge transformation $A' = \Omega^{-1}A\Omega + \Omega^{-1}d\Omega$, where Ω is a G-valued function on $\mathcal{R}^4$. The self-duality equations are $*F = F$ where $*$ is the Hodge duality operator. These equations form a set of coupled first-order nonlinear partial differential equations for A_a. They are underdetermined (fewer equations than unknowns), but this underdeterminacy can be removed by imposing a 'gauge condition' such as $A_0 = 0$. The self-duality equations are invariant under gauge transformations. The self-duality equations are completely solvable as a consequence of the twistor correspondence. The equations possess the Painlevé property. Many other integrable partial differential equations which also have the Painlevé property can be derived from the self dual Yang Mills equation. Finally, let us discuss an interesting connection of the self-dual Yang Mills equation and the Yang Mills equation. From the self-dual Yang Mills equation we find, after reduction, the system of ordinary differential equations $du_1/dt = u_2u_3$, $du_2/dt = u_1u_3$, $du_3/dt = u_1u_2$ When we differentiate this system with respect to t and insert it into the new second order equations we arrive at

$$\frac{d^2u_1}{dt^2} = u_1(u_2^2+u_3^2), \quad \frac{d^2u_2}{dt^2} = u_2(u_1^2+u_3^2), \quad \frac{d^2u_3}{dt^2} = u_3(u_1^2+u_2^2).$$

This system does not have the Painlevé property. It is not integrable. From the Yang-Mills equation $D(*F) = 0$ we find, after reduction, the nonintegrable system

$$\frac{d^2u_1}{dt^2} = -u_1(u_2^2+u_3^2), \quad \frac{d^2u_2}{dt^2} = -u_2(u_1^2+u_3^2)$$
$$\frac{d^2u_3}{dt^2} = -u_3(u_1^2+u_2^2).$$

This system is nonintegrable and can be derived from the Hamilton function $H(\mathrm{u},\dot{\mathrm{u}}) = (\dot{u}_1^2+\dot{u}_2^2+\dot{u}_3^2)/2+(u_1^2u_2^2+u_1^2u_3^2+u_2^2u_3^2)/2$. It shows chaotic behaviour. Furthermore, we find that it does not pass the Painlevé test. The resonances are given by -1, 1 (twofold), 2 (twofold) and 4. Studying the behaviour at the resonances we find a logarithmic psi-series. The two systems are equivalent up to the sign on the right hand side. Since we consider the system in the complex domain for the Painlevé test we can find that the two systems are related via the transformation $t \to it$.

REFERENCES

1. Weiss J., Tabor M. and Carnevale G., *J. Math. Phys.*, 24, 522 (1983).

2. Ward R. S., *Nonlinearity* 1, 671 (1988).

3. Steeb W.-H. and Euler N., *Nonlinear Evolution Equations and Painlevé Test*, World Scientific, Singapore, 1988.

4. Steeb W.-H. *Problems in Mathematical Physics, Volume II: Advanced Problems*, Bibliographisches Institut, Mannheim, 1990.

5. Clarkson P. A., *IMA Journal of Applied Mathematics* 44, 27 (1990).

6. Euler N., Steeb W.-H. and Cyrus K., *Physica Scripta* 41, 289 (1990).

SOME NUMERICAL METHODS FOR LOW DIMENSIONAL CHAOTIC DYNAMICAL SYSTEMS

by Helena E. Nusse

Fac. der Economische Wetenschappen R.U. Groningen,
Postbus 800, NL-9700 AV Groningen. The Netherlands

Studying dynamical systems, one often observes transient chaotic behavior. Famous examples are the Hénon map, the forced damped pendulum, the forced Duffing equation and the Lorenz equations. Let F be a differentiable, invertible map from the n-dimensional phase space to itself, such that the derivatives of F and its inverse are continuous. Let R be a transient region, that is, R is an open and bounded set in the phase space that contains no attractor. The stable set S(R) is the set of points (in R) which stay in R for all time under forward iteration of F; we refer to R\S(R), the complement of the stable set S(R) in the transient region R, as the transient set. The invariant set Inv(R) of F in R is the set of points in R which stay in R for all time under forward and backward iteration of F, and we assume that Inv(R) is nonempty. We study transient regions in cases where the trajectory through almost every initial point eventually leaves the region. We are looking for trajectories that stay in the region R as long as we wish to compute them. A first question is "*Find a chaotic trajectory numerically that remains in the region R for an arbitrarily long period of time.*"

A point p in S(R) is accessible from an open set V if there is a continuous curve K ending at p such that K\{p} is in V. We first investigate the case where V is the transient set R\S(R). A second question is: "*Given a line segment J that crosses S(R) transversally. Describe a procedure for finding a numerical trajectory on the stable set S(R) that starts on J and which is accessible from transient set R\S(R).*"

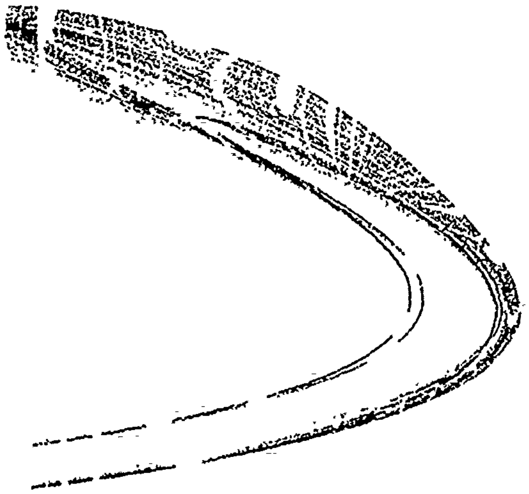
Nonlinear dynamical systems often have more than one attractor. The basin boundary is the set of all points for which each open neighborhood contains points of at least two different domains of attraction. A generalized attractor A is the union of finitely many attractors, and we define basin{A} to be the interior of the closure of the domain of attraction of A. We now assume for the transient region R that there exist 2 generalized attractors A and B, and each point in R that escapes from R under iteration of the map F is either in basin{A} or in basin{B}. The stable set S(R) might contain points on the basin boundary that are accessible from basin{A} (or basin{B}) but not accessible from the transient set R\S(R). Hence, the ASST method for finding accessible points on S(R) is, generally speaking, not a procedure for finding accessible points on the basin boundary. A third question is: "*Given a line segment J that has one end point in basin{A} and the other end point in basin{B}. Describe a procedure for finding a numerical trajectory on the basin boundary that starts on J and which is accessible from basin{A}.*"

The numerical methods ("straddle methods") described below have been developed in colloboration with J.A. Yorke. We restrict our attention to cases in which there is only one positive Lyapunov exponent.

For the straddle methods below, we use the notions of "escape time" and "ε-refinement". The escape time T(x) of a point x in the transient region R under the map F is the minimum value n such that $F^n(x)$ is not in R; the escape time T(x) is infinity if $F^n(x)$ is in R for all n. For {x,y} on a line segment J we always assume for convenience that the ordering on J is such that we may write $x < y$; we denote [x,y] for the segment joining x and y, and $|x-y|$ for the distance of x and y. Let {a,b} be a pair of points on J. For every $\varepsilon > 0$, an ε-refinement of {a,b} is a finite set of points $a = g_0 < g_1 < \ldots < g_N = b$ in [a,b] such that $\frac{\varepsilon}{2}\cdot|a-b| \le |g_k-g_{k+1}| \le \varepsilon\cdot|a-b|$ for all k, $0 \le k \le N-1$.

STRADDLE METHODS. Straddle methods involve a refinement procedure in which 2 points on a curve segment are replaced by 2 new points In some cases the points have different roles. Usually each of the refinement procedures takes a pair of points and returns a pair of points; such a returned pair is on the line segment joining the two points of the original pair. The distance between the two points in the returned pair is smaller than the distance between the points of the original pair. Straddle methods consist of applying the refinement procedure repeatedly until the points in the resulting pair are less than some specified distance σ apart, say $\sigma = 10^{-8}$. If the points in the original pair are already less than σ apart, then no refinement is carried out. Next apply the dynamics; that is, apply the map F to each of the 2 points of the resulting pair, giving a new pair.

The basic numerical method takes a pair $\{a_n,b_n\}$ which is separated by at most a distance σ, and applies the map F to each of the points of this pair. If the new pair $\{F(a_n),F(b_n)\}$ is separated by less than σ, then it is denoted $\{a_{n+1},b_{n+1}\}$, and otherwise the refinement procedure is applied repeatedly until a pair with separation at most σ is obtained, and it is called $\{a_{n+1},b_{n+1}\}$. However, in order to produce the first pair $\{a_0,b_0\}$, the method starts by applying the refinement procedure on the given pair $\{a,b\}$, whose points are presumably more than σ apart. Writing I_n or $[a_n,b_n]$ for the line segment from a_n to b_n, and to the precision of the computer we usually have $I_{n+1} \subset F(I_n)$. We call the sequence of tiny straight line segments $\{I_n\}_{n\geq 0}$ a straddle trajectory.



SST METHOD. The "saddle dynamic restraint problem" is to describe a numerical method for finding a trajectory that remains in a specified transient region for an arbitrarily long period of time. First, we describe the refinement procedure that is involved in the current straddle method. Let $\{a,b\}$ be a pair such that $[a,b]$ intersects $S(R)$ transversally.

Let (α,γ,β) be a triple on $[a,b]$. We call (α,γ,β) an Interior Maximum triple if both $T_R(\gamma) > T_R(\alpha)$ and $T_R(\gamma) > T_R(\beta)\}$; we call (α,γ,β) a PIM triple if (α,γ,β) is an Interior Maximum triple and $|\beta-\alpha| < |b-a|$. Assume that (α,γ,β) is an Interior Maximum triple, and let P be any ε-refinement of $\{\alpha,\beta\}$ including γ. The procedure that selects in the refinement P any PIM triple $(\alpha^*,\gamma^*,\beta^*)$ is called a PIM triple (refinement) procedure.

The solution to the "saddle dynamic restraint problem" is the straddle trajectory using the PIM triple procedure. We call the sequence of tiny straight line segments $\{I_n\}_{n\geq 0}$ a saddle straddle trajectory or SST trajectory, and we call the straddle method that generates the SST trajectory $\{I_n\}_{n\geq 0}$, the SST method. Notice that each tiny line segment in an SST trajectory straddles a piece of a (chaotic) saddle. An SST trajectory typically resembles (after a few iterates) a basic set in the (chaotic) saddle.

ASST METHOD. The "accessible saddle dynamic restraint problem" is to describe a numerical method for finding a trajectory on the stable set S(R) that is accessible from the transient set R\S(R). The refinement procedure that is involved in the current straddle method is a PIM triple (refinement) procedure in which a PIM triple $(\alpha^*,\gamma^*,\beta^*)$ is selected from the ε-refinement P of the Interior Maximum triple (a,c,b) such that $[a,a^*]$ is in the transient set. This refinement procedure is called the Accessible PIM triple (refinement) procedure. The solution to the "accessible saddle dynamic restraint problem" is the straddle trajectory using the Accessible PIM triple procedure. We call the straddle trajectory $\{I_n\}_{n\geq 0}$ an accessible saddle straddle trajectory or ASST trajectory, and we call the straddle method that generates the ASST trajectory $\{I_n\}_{n\geq 0}$, the ASST method. An ASST trajectory typically resembles (after a few iterates) a subset of the nonwandering points in R which are accessible from the set R\S(R).

ABST METHOD. The "accessible basin boundary dynamic problem" is to describe a numerical method for finding a trajectory on the basin boundary that is accessible from basin{A}. The refinement procedure in the current straddle method that generates a proper straddle pair. Let {a,b} be a straddle pair such that a is in basin{A} and b is in basin{B}, and [a,b] intersects the stable set S(R) transversally. Let P be any $\frac{\varepsilon}{3}$-refinement of {a,b}. In the unique proper straddle pair $\{a^*,b^*\}$ from P the point b^* is the leftmost point of P that is in basin{B}, and a^* depends on the grid consisting of b^* and all the points in P to the left of b^*. The solution to the "accessible basin boundary dynamic restraint problem" is the straddle trajectory using this refinement procedure. We call the straddle trajectory $\{I_n\}_{n\geq 0}$ an accessible basin boundary straddle trajectory or ABST trajectory, and we call the straddle method that generates this trajectory, the ABST method.

In this work we study the planar cubic map $Ho(x,y) = (ax - x^3 + by, x)$ and the 3-dimensional Hénon-like map $HZ(x,y,z) = (1 + y - zx^2, bx, z - 0.5 + ax^2)$.

REFERENCES

H.E. Nusse and J.A. Yorke. A procedure for finding numerical trajectories on chaotic saddles. Physica D 36 (1989), 137-156.

H.E. Nusse and J.A. Yorke. Analysis of a procedure for finding numerical trajectories close to chaotic saddle hyperbolic sets. To appear in Ergodic Theory and Dynamical Systems (1991).

H.E. Nusse and J.A. Yorke. A numerical procedure for finding accessible trajectories on basin boundaries. To appear in Nonlinearity (1991).

PHYSIC-LIKE MATHEMATICS OF CHAOS AND ERGODIC CRITICALITY IN FOUR DIMENSIONS

M.S. El Naschie
Sibley School of Mechanical & Aerospace Engineering
Cornell University
112 Upson Hall
Ithaca, N.Y. 14853, U.S.A.

Abstract

The present discussion attempts to show some mathematical connections be.ween chaotic dynamics, fractal sets and dimensionality which may have relevance to physical systems.

1. Introduction

At least since the spacetime of relativity four dimensions have occupied a special place in physics. Modern nonlinear science has reinforced this situation. O. Roessler for instance has repeatedly drawn attention to four dimensional sets as the frontiers of new phenomina associated with wrinkled and hairly attractors [3]. Ruelle, Takens and Newhouse envisages chaos as a sequence of a finite number of Hopf bifurcations leading to a totally unstable torous in four dimensions [4]. In fact one of the most important discoveries in topology implies that four dimensions are more complicated than any lower dimension and surprisingly even higher dimension [5]. To motivate our approach consider a recent result [2] where it was found that haemoglobin has a surface fractal dimension $d = 2.4$. Now this is more profound than finding for instance a two dimensional object with a fractal dimension less than two because somehow holes could account for the reduction. Here however they are insisting that it is a surface.Consequently the 0.4 may be attribu - ted to "negative" holes in the surface. It must be a very rugged surface full with little mountains trying in a sense to jump out into the surrounding three dimensions. Should we ever encounter a surface which has a fractal dimension $d > 3$, this would be even more radical. In a "facon de parler" such a surface wants to reach out of the embedding three dimensional Eucledian manifold. That way $d=3$ may be regarded as a "critical" dimension for this "strange" surface. Generalizing to n dimensions we could say that whenever the fractal dimension of an object and the dimension of the hosting manifold becomes equal a critical state in the preceding sense is reached. For reasons which we are about to make clear here, we will term this critical state quasi ergodic criticality. In what follows we would like to show that under a fairly reasonable assumption one can conclude that four dimensions marks a special point in chaotic dynamics.

2. Cantor-like sets and critical ergodicity

The starting point of our analysis is the generally accepted realization that fractals [6] are the carriers of complex strange behaviour. Second and without going into detail, we follow Yorke's conjecture that single Cantor sets are some how the back bone of all strange behaviour [7]. To that we add what we intuitively feel as evident namely that in one dimension it is extremely hard to think of any simpler fractal set than Cantor's middle third set [2] with $d_c = \text{Log } 2/\text{Log } 3$. If we can accept all this, which is not particularly easy to justify we can claim that in four dimensional phase space a strange set will typically have a Cantor-like fractal dimension $d_c \cong 4$. This result is reached using the following scaling argument. The idea is to find the equivalent to a triadic Cantor set in two dimensions. Such a set should be triadic Cantorian in every conceivable direction. It cannot therefore be the Cartesian product of two such sets, $d_c = \text{Log } 4/\text{Log } 2$ nor a Cantor target $d_c = 1 + \text{Log } 2/\text{Log } 3$. However we know that a unit area A of an Euclidian manifold is given by $A = (1)(1) = 1$ and consequently a corresponding or quasi area of a Cantor set is $A_c = (d_c)\ (d_c)$. It follows then that in order to normalize A_c it must be multiplied by the normalization factor $\wp_2 = (A/A_c)_2$. By analogy in n dimensions we would have $\wp_n = (A/A_c)_n$. Denoting the n-th Cantor-like fractal dimension in n dimensional space by $d_c^{(n)}$ and the dimension of the corresponding Euclidian space in n dimensions by $d_E^{(n)} = n$ it follows then that

$$d_c^{(n)} = \wp_n d_c = \frac{d_c}{(d_c)^n} = \left(\frac{1}{d_c}\right)^{n-1} = (d_s)^{n-1}$$

where d_s is termed the escalation factor. This is the set which we are looking for and the result is now evaluated for $d_c = \text{Log}2/\text{Log } 3$ in table 1. Note that d_s could be equally interpreted as the Floquet multiplier of a discrete map

$$d_{(m+1)} = d_{(m)} \frac{1}{d^{(o)}}$$

where $n = m + 1$, $\quad d^{(o)} = d_c^{(o)}$ and

$$d_s = \partial(d_{(m+1)})/\partial(d_{(m)})$$

	$d_E^{(n)}$	$d_c^{(n)}$
Basic assumption	0	$d_c^{(o)} = 0.63092$
Normality	1	1
Results	2	1.58496
	3	2.51210
	4	3.98159
	5	6.31067
	6	10.00218
	7	15.85309
	8	25.12655

- TABLE 1 -

There are a few interesting observations here. First $d_c^{(n+1)} / d_c^{(n)} = d_s$ is the fractal dimension of the Serpenski gasket which is the prototype of fractal lattices with infinite hierachy of loops. Second for all $n < 4$ we have $n > d_c^{(n)}$ while for $n > 4$ we have $d_c^{(n)} >> n$. Only at $n=4$ we have a Cantor-like structure which comes very near to a space filling set. The two dimensional geometrical analogue of this is the peano curve which is ergodic and shares a few properties with fat fractals [8]. We may say therefore that at n=4 the set is almost ergodic. The third observation is that for any three successive dimension$d^{(n)} \approx d^{(n-1)} + d^{(n-2)}$ This is strongly reminiscent of the Fibonacci numbers [2] and the corresponding dimension will be termed the Fibonacci fractal dimension. Should we insist that $d^{(n)} = d^{(n-1)} + d^{(n-2)}$ then we find that at n=4 the corresponding Cantor-like dimension is d_c = 4.23606 while the Serpenskı gasket [8] is replaced by $d_S = 1/\phi$ where ϕ is the Golden mean [2]. In fact our table number 1 becomes identical to the table calculated by Cook [1] for Botticelli's venus. The next step is of course the obvious thing to do. We determine the escalation value d_S corresponding to exact critical equality of $d_c^{(n)}$ and n in four dimensions. This is an elementary application of our formula relating $d_c^{(n)}$ to n. This way one finds

$$\left(\frac{1}{d_c^{(n)}}\right)^{n-1} = \left(d_s\right)^{n-1} = n \;;\; \left(\frac{1}{d_c^{(n)}}\right)^{4-1} = d_s^{4-1} \;;\; d_s = \sqrt[3]{4} = 1.587$$

This is very close to the Serpenski gasket [8] d_c = 1.58496. Now a single Cantor set is easily made to have any fractal dimension between one and zero. Withın this range it is now interestıng to consider the consequence of having taken a Cantor set with Hausdorf dimension d_c = Log 2/Log 4. In this case d_c = 0.5 seems to be a distinct value between one and zero which might be regarded naively as the most "fractal" value in this unit interval. It is also the correlation dimension found for period doubling chaos in the one dimensional logistic map as well as the probabılıty descrıbing the random behaviour of the tent map. It is an elementary matter to show, using the same previous formula, that d_c = 2. The critical state thus shifts from n = 4 to n = 2.

This is however another way of viewing the "critical" ergodic state n=4. It is also related to quasi periodically forced horse shoe maps displaying peano-like dynamics [10]. Finally let us consider the implication of shifting criticality in the present ergodic sense to n=3. This clearly implies an escalation factor d = $\sqrt{3}$ = 1.732050. Notice that in this case $d_c^{(2)}$ = 1.73205 is indeed a value found frequently in two dimensional Poincare maps of dynamic system as well as numerous fractal objects found in nature [8]. The role of multifractals as well as fractal sets made up of the union of different fractal subsets in developing more accurate mathematical model will not be discussed here.

3. Concluding Remarks

Looking back at table 1. One may be lead to speculate if fully developed turbulence has a fractal dimension d ≅ 6.3 and that five dimensional phase space is required to study this phenomina. This would be for instance a nonlinearly oscillating set described by a phase space x , $\dot{x}$ and $\grave{x}$ representing temporal and spacial oscilation of a state variable x. In addition we need a spacial fluctuation ω_x and a temporal fluctuation ω_t as forcing frequencies. This makes them indeed five variables. Another worthwhile observation is that the Fibonacci fractal dimension $d_F^{(3)}$ = 1 + 1/(Log 2/Log 3) = 2.58496 is identical to d_S = 1/(Log 2/Log 6) where Log 2/Log 6 is clearly a reasonable measure of the fractal dimension at period 3 chaos of a Feigenbaum cuscad. Note also that d_c = Log 2/Log 6 ≅ 0.387 is very close to the smallest value found for period 3 chaos of the logistic map [9,11] (d ≅ 0.378).

It is, of course important to appreciate the role of the proximity of some rational and irrational numbers such as 3/4, π/5, and $\phi = (1 + \sqrt{5})/2$ and d_c = Log 2/Log 3, in arriving at some of the preceding conclusions.

References

1. Cook, T.A. The curves of life. Reprinted by Dover Publications, New York (1979). Originally published by Constable and Company, London (1914).

2. Stewart, I. Does God play dice? Penguin, London (1989).

3. Rossler, O.E., Hudson, J.L., Klein M. and Mira, C. Self similar basin in continuous systems. In "Nonlinear Dynamics in Engineering Systems". Editor W.Schiehlen, p.265-273,Springer (1990).

4. Ruelle, D. and Takens, F. On the nature of turbulence. Commun. Math. Phys., 20, 167 (1971).

5. Stewart, I. The problems of mathematics. Oxford University Press (1987).

6. Becker, K.H. and Dorfler, M. Dynamical systems. Cambridge Press, English translation by I. Stewart. (1989).

7. Eubank, S. and Farmer, D. Introduction to chaos and randomness. In "1989 lectures in complex systems". Editor E. Jen, pp. 75-190, Addison Wesley, Redwood City (1989).

8. Vicsek, T. Fractal growth phenomena. World Scientific, Singapore (1989).

9. Grossmann, S. Selbstaehnlichkeit, Das Strukturgesetz im und vor dem Chaos. In "Ordnung und Chaos". Editor W. Gerok, pp. 101-122, S. Hirzel Wissenschaftlıcher Verlag, Stuttgart (1989).

10. Kapitaniak, T. On strange nonchaotıc attractors and their dimensions, Chaos, Solitons and Fractals (a new journal by Pergamon Press) Vol 1, number 1 (1991).

11. El Naschie, M.S. Stress, Stability and Chaos. McGraw Hill, London (1990).

APERIODICITY AND SENSITIVE DEPENDENCE ON INITIAL CONDITIONS IN QUASIPERIODICALLY FORCED SYSTEMS

T. Kapitaniak , Department of Applied Mathematical
Studies and Center for Nonlinear
Studies, University of Leeds,
Leeds Ls2 9JT, U.K.

and

M. S. El Naschie, Sibley School of Mechanical
and Aerospace Engineering,
Cornell University, Ithaca,
NY 14853-7501, USA.

ABSTRACT

Typical similarities and differences between strange chaotic and nonchaotic attractors in deterministic systems and random behaviour are discussed. It has been shown that based on a single time series it is impossible to distinguish between these types of behaviour even using Lyapunov exponents technique.

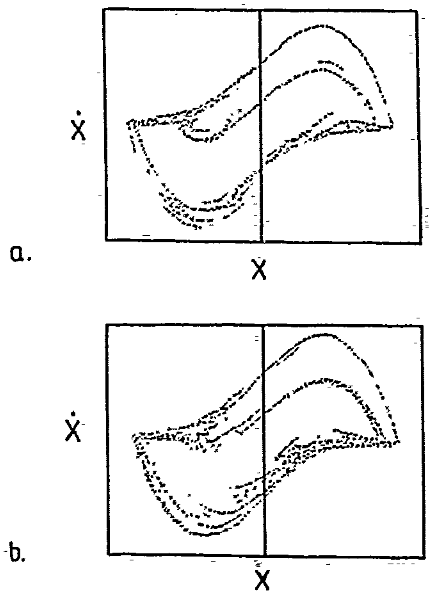


Fig. 1. The Poincare maps of the quasiperiodically forced eq. (1): a=5.0 , d=5.0 , Ω=√2 + 1.05 ; (a) strange chaotic attractor ω = 0.002, the largest nonzero Lyapunov exponent λ=0.1183, (b) strange nonchaotic attractor ω=0.006, λ=-0.1213.

In last decade attention has been given to a class of dissipative dynamical systems that typically exhibit strange behaviour [1-3]. Such behaviour has been found in numerical experiments [3,4] as well as in experimental systems [5,6].

Recently two classes of strange attractors have been distinguished:

(a) a strange chaotic attractor - one which is geometrically, strange' i.e. the attractor is neither a finite set of points nor it is piecewise differentiable and one for which typical orbits have positive Lyapunov exponents

(b) a strange nonchaotic attractor - one which is also geometrically 'strange' but for which typical nearby orbits do not diverge exponentially with time [7-9, 15-21].

Strange nonchaotic attractors have been found to be typical for quasiperiodically forced systems [7,18]. Recently they have been also observed in experimental system [21].

Although one may doubt that these are periodic or quasiperiodic orbits with sufficiently long period but even in this case period is longer than any reasonable observation and that is why their name is justified.

As both types of strange attractors look very similar (compare for example Poincare maps of Figure 1(a) and (b)), the value of Lyapunov exponents seems to be the only quality which allows us to distinguish these classes.

In this paper we present some numerical experiments showing that it is impossible to distinguish between strange chaotic and nonchaotic deterministic behaviour basing on a single time series.

For systems which equations of motion are explicitly known and the linearized equations exist there is a straightforward technique [10,11] for computing a complete Lyapunov spectrum.

For most of the experimental systems the equations of motion are not known [12] or are in the form for which the linearized equations do not exist [6,13]. In this case Lyapunov exponents are estimated based on the monitored long - term time series. First the attractor is reconstructed by the well-known technique with delay coordinates [14]. Our reconstructed attractor though defined by a single trajectory can provide points that may be considered to lie on different trajectories. It has been shown that in many cases this attractor has got Lyapunov spectrum identical to that of the original attractor [10].

The technique of estimation of Lyapunov exponents based on the reconstructed attractor gives good results when we have at least one positive Lyapunov exponent in the spectrum.

In what follows we consider the quasiperiodically forced Van-der-Pol's equation

$$\ddot{x} - a(1 - x^2)\dot{x} + x = d\cos\omega t\cos\Omega t \qquad (1)$$

For eq. (1) linearized equations exist and we can compute Lyapunov exponents directly from the formula :

$$\lambda = \lim_{t\to\infty} \frac{d(t)}{d(0)} \qquad (2)$$

where: $d = \sqrt{y^2 + \dot{y}^2}$, while y is a solution of a linearized equation.

In these cases we have computed Lyapunov exponents twice from the formula (2) and from time series based on Wolf et al. algorithm [10]. In numerical similations the four order Runge-Kutta method with time step T/200, where T=2Π/ω has been used. Strange nonchaotic attractors have been observed up to $T=10^8$. The comparison of these results is shown in Figure 2.

From Figure 2 one finds that the calculation of the Lyapunov exponents from the explicitly known differentiable equation allows to distinguish between strange chaotic and nonchaotic attractors. This distinction cannot

a	5.0	5.0	5.0	5.0
d	5.0	5.0	5.0	5.0
ω	0.006	0.007	0.003	0.002
Ω	$\sqrt{2}+1.05$	$\sqrt{2}+1.05$	$\sqrt{2}+1.05$	$\sqrt{2}+1.05$
λ_{max}				
formula (2)	-0.1213	-0.2834	0.1426	0.1468
time series	0.0845	0.0684	0.1183	0.1232
Type of attractor	strange nonchaotic		strange chaotic	

Fig. 2. The comparison of the values of Lyapunov exponents λ_{max} computed from formula (2) and estimated from time series.

be followed based on the Lyapunov exponents estimated from a single time series, as by this method we obtained positive values of Lyapunov exponents not only in the case of strange chaotic attractors but for strange nonchaotic attractors as well.

This result may look quite surprising but it is justified when we follow the method of Lyapunov exponents estimation from the attractor reconstructed from single time series. If a time series is irregular (not periodic, quasiperiodic) it is not distinctive from chaotic one and the reconstructed attractor has got the complicated geometry. Estimating Lyapunov exponents from this attractor we have to obtain a positive values for both strange chaotic and nonchaotic attractors, as the whole procedure explores the aperiodicity of time series and not the explicite dependence on initial conditions. As other methods of estimating Lyapunov exponents from time series [23-26] are also based on the same method of attractor reconstruction it seems that using them similar results are very likely.

The possiblity of having a system showing strange behaviour without sensitive dependence on initial conditions should not be overlooked. It seems that more care will have to be given in applying theprocedure of estimation of Lyapunov exponents from time series to experimental data. The general conclusion that it imply can be misleading, as there are systems for which distinction between strange chaotic, strange nonchaotic behaviour is impossible based on a single time series.

REFERENCES

1 H. G. Schuster, Deterministic Chaos, VCH, Weinheim (1988)

2 B. L. Hao, Chaos, World Scientific, Singapore (1990)

3. U. Parlitz and W. Lauterborn, Phys. Lett., 107A, 351 (1985)

4. Y. Ueda , J. Stat. Phys., 20 , 181 (1979)

5. G. Qin et al., Phys. Lett. 141A, 412 (1989)

6. B. F. Feeny and F.C. Moon , Phys. Lett. 141A, 412 (1989)

7. F. J. Romeiras and E. Ott, Phys. Rev. 35A, 4404 (1987)

8. T. Kapitaniak et al., J. Phys A, 23, (1990)

9. M.S. El Naschie and T. Kapitaniak , Phys. Lett. A, (1990)

10. A. Wolf et al., Physica 16D, 285 (1985)

11. T. Kapitaniak, Chaotic Oscillations in Mechanical Systems, Manchester University Press (1990)

12. J.-C. Roux et al., Physica, 8D, 2, (1982)

13. K. Poop and P. Stelter, Nonlinear Oscillations of Structures Induced by Dry Friction . In: W. Schiehlen (ed.) , Nonlinear Dynamics in Engineering Systems, Springer (1990)

14. N.H. Packard et al., Phys. Rev. Lett., 45, 712 (1980)

15. M. Ding et al., Phys. Lett. 137A, 167, (1989)

16. C. Grebogi et al., Physica 13D, 261(1985)

17. A. Bondeson et al., Phys. Rev. Lett.,55, 2103 (1985)

18. F. J. Romeiras et al.,Physica, 26D, 277, (1987)

19. M. Ding et al., Phys. Rev. A39, 2593 (1989)

20. T. Kapitaniak, and J. Wojewoda, J. Sound Vibration, 138, 162 (1990)

21. W.L. Ditto et al., Phys. Rev. Lett., 65, 533 (1990)

22. E. Stone, Phys. Lett., 148A, 434 (1990)

23. J.-P. Eckmann and D. Ruelle, Rev. Mod. Phys. 57, 617 (1985)

24. M. Sano and Y. Sawada, Phys. Rev. Lett. 55, 1082 (1985)

25. J. -P. Eckmann et al., Phys. Rev. A34, 4971 (1986)

26. K. Briggs, Phys. Lett., 151A, 27 (1990)

INFLUENCE OF FRICTION ON THE CHAOTIC DYNAMICS IN COUPLED OSCILLATORS.

JAN AWREJCEWICZ
The University of Tokyo
Department of Mechanical Engineering
7-3-1 Hongo, Bunkyo-ku,
Tokyo, JAPAN

AND

WOLF D. REINHARDT
Technical University
Department of Mechanical Engineering
Langer Kamp 19B,
3300 Braunschweig, GERMANY

Abstract - A route to chaos in the system with dry friction is analyzed. In spite of the complexity of the system, a similar transition to that discovered in the two-well potential anharmonic oscillator is described and illustrated. Dry friction weakens the chaotic dynamics and induces the occurrence of stick and slip transitions during the chaotic wandering of the trajectory in four dimensional phase space.

1. INTRODUCTION

The aim of this paper is to show the "qualitative universal" transition to chaos in a certain subclass of sinusoidally-driven nonlinear oscillators, i. e., systems with a two-well potential. The question of interest is whether or not the scenarios leading to chaotic orbits discovered in simple uncoupled oscillators are likely also hold for much more complicated systems, such as coupled nonlinear sinusoidally driven oscillators. Simulation experiments show that the potential has two wells, and that chaotic dynamics will obtain, in which each of the oscillators jumps between two wells in an unpredictable way.

2. THE SYSTEM

We consider a system of two coupled mechanical oscillators, both of which are externally driven. The governing equations are

$$m_1\ddot{x}_1+(C_3-C_1)\dot{x}_1-C_3\dot{x}_2+C_2x_1^2\dot{x}_1+(k_1+k_3)x_1$$

$$-k_3x_2+k_2x_1^3 \ +\mu m_1 g\,sgn(\dot{x}_1)= \ q_1\cos(\omega_1 t+\phi).$$

$$m_2\ddot{x}_2+(C_3-C_4)x_2-C_3x_1+C_5x_2^2x_1+(k_3+k_4)x_2$$

$$-k_3x_1+k_5x_2^3 \ = \ q_2\cos(\omega_2 t), \qquad (1)$$

where m_1 and m_2 are the masses of the oscillators, $C_1 - C_5$ and $k_1 - k_5$ are damping and stiffness coefficients, respectively q_1 and q_2 are the amplitudes of the exciting forces with corresponding frequencies ω_1 and ω_2, and ϕ denotes a phase shift between the exciting forces.

In nondimensional form we have

$$\xi_1''+(\alpha_3-\alpha_1)\xi_1'-\alpha_3(KM^{-1})^{0.5}\xi_2'+\gamma_1\xi_1^2\xi_1'+(\kappa_1+\kappa_3)\xi_1$$

$$-\kappa_3(KM^{-1})^{0.5}\xi_2+\xi_1^3+R\,sgn(\xi_1') \ = \ B_1\cos(\tau+\phi).$$

$$\xi_2''+M(\alpha_3-\alpha_4)\xi_2'-M\alpha_3(MK^{-1})^{0.5}\xi_1'+\gamma_2K\xi_2^2\xi_2'+M(\kappa_3+\kappa_4)\xi_2$$

$$-M\kappa_3(MK^{-1})^{0.5}\xi_1+\xi_2^3 \ = \ M^{1.5}K^{-0.5}B_2\cos(\nu\tau). \quad (2)$$

where

$$\tau=\omega_1 t, \xi_1=(\omega_1 m_1^{0.5})^{-1}k_2^{0.5}x_1, \xi_2=(\omega_1 m_2^{0.5})^{-1}k_5^{0.5}x_2,$$

$$M=m_1m_2^{-1}, \quad K=k_2k_5^{-1}, \quad \nu=\omega_2\omega_1^{-1}.$$

$$B_1=q_1\omega_1^{-3}m_1^{-1.5}k_2^{0.5}, B_2=q_2\omega_1^{-3}m_1^{-1.5}k_2^{0.5}, \alpha_1=C_1(m_1\omega_1)^{-1},$$

$$\alpha_3=C_3(m_1\omega_1)^{-1}, \alpha_4=C_4(m_1\omega_1)^{-1}, \kappa_1=k_1m_1^{-1}\omega_1^{-2},$$

$$\gamma_1=\omega_1C_2k_2^{-1}, \quad \gamma_2=\omega_1C_4k_2^{-1}, \quad \kappa_3=k_3m_1^{-1}\omega_1^{-2},$$

$$\kappa_4=k_4m_1^{-1}\omega_1^{-2}, \quad R=\mu g\omega_1^{-3}k_2^{0.5}m_1^{-0.5}. \qquad (3)$$

Using the transformations (3), the nineteen parameters of equations (1) are reduced to fourteen parameters in (2).

Such a general system has been investigated earlier by the authors [1-3] using a systematical numerical approach. Transitions between quasiperiodic, strange chaotic and strange non-chaotic attractors have been reported as well as some special chaotic dynamics has been discussed and illustrated. Here an attention is focused on the influence of friction of the chaotic dynamics on the mentioned above system.

3. NUMERICAL ANALYSIS

We define: $F_{st}=\mu m_1 g$,

$$F=(k_1+k_3)x_1-k_3x_2+k_2x_1^3-c_3x_2-q_1\cos\omega_1 t. \qquad (4)$$

When $\dot{x}_1=0$ and $|F|<|F_{st}|$, the first oscillator is in a stick state. During the transition from a slip-state to a stick-state, an acceleration jump occurs. During an exit from a stick-state to a slip-state the acceleration is continuous, but a jump in the third derivative of the displacement appears. The velocity in both cases is always continuous. During "the transition over a stick area", but without sticking, the acceleration jump also occurs. When a regular transition, however, from a slip-state to a stick takes place with $|F|<|(F_{st})_{max}|$, the one eigenvalue of the Jacobi matrix is equal to zero. When $|F|=|(F_{st})_{max}|$ and the acceleration jump does not appear, the Jacobi matrix is not singular. The following equations govern the dynamics in the stick-state:

$$\xi_1=0,$$

$$\xi_2''+M(\alpha_3-\alpha_4)\xi_2' + \gamma_2K\xi_2^2\xi_2'+M(\kappa_3+\kappa_4)\xi_2$$

$$-M^{1.5}\kappa_3K^{-0.5}\xi_1+\xi_2^3 \ = \ M^{1.5}K^{-0.5}B_2\cos\nu t. \qquad (5)$$

with the following transition-condition

$$R > |(\kappa_1+\kappa_3)\xi_1-\kappa_3(K/M)^{0.5}\xi_2+\xi_1^3$$
$$-\alpha_3(K/M)^{0.5}\xi_2'-B_1\cos\tau|. \qquad (6)$$

We consider the behavior of the system (2) for the following fixed parameters: $\nu=M=K=1.0$, $\kappa_1=\kappa_4=-0.816326$, $\phi=0.0$, $\gamma_1=\gamma_2=0.3$, $B_1=0.05$, $B_2=0.2$, and for two values of friction R.

Example 1 (R=0.05).

We take $\alpha_1=\alpha_4=0.01$ and $\alpha_3=\kappa_3=0.3$. For these parameters and without friction (R=0), the system exhibits intermittent chaos. Friction dampens the chaotic dynamics of the orbits and for R=0.05 we find a quasiperiodic attractor. In the neighborhood of these parameters (for $\alpha_1=\alpha_4=0.05$ and $\alpha_3=\kappa_3=0.3$), a periodic attractor is found. An increase in α_1 ($\alpha_1=\alpha_4$) results in an increase in the magnitude of the self-excited oscillations. The periodic orbit grows and finally leads to intersections of the stable and unstable manifolds and a trajectory starts to wander in an unpredictable way between two potential wells. This situation is illustrated for $\alpha_1=\alpha_4=0.2$ in Fig.1.

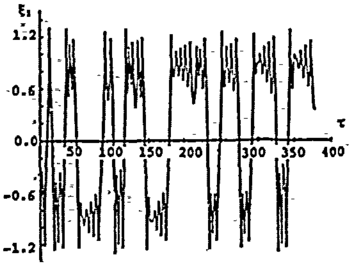


Fig.1. Time history of a strange chaotic attractor (R=0.05).

Example 2 (R=0.1).

In the second example we analyze the influence of the coupling between two oscillators. The numerical calculations have been carried out for the same parameters as in Example 1 and additionally for R=0.1 and $\alpha_1=\alpha_4=0.2$. When two oscillators are strongly coupled ($\alpha_3=2.0$, $\kappa_3=0.3$), a periodic orbit is found. This orbit lies to the right of the origin. However, to the left of the origin there is also another small periodic orbit. These two orbits lie in two isolated potential wells. Decreasing α_3 causes the trajectory to move from the potential well and start to wander between the two potential wells (Fig.2). The escape, however, from one of the wells to the other is rather rare. The possibility of it occurring increases with a further decrease in α_3. For example, for $\alpha_3=0.3$, one of the projections of the Poincaré map shows a very complicated dynamics (Fig.3).

In order to understand how two oscillators move in a chaotic manner, two time histories (for relatively long time intervals of the same chaotic attractors) are presented in Fig.4. In this figure one can also observe stick states. These states correspond the very short horizontal parts of $\xi_1(\tau)$.

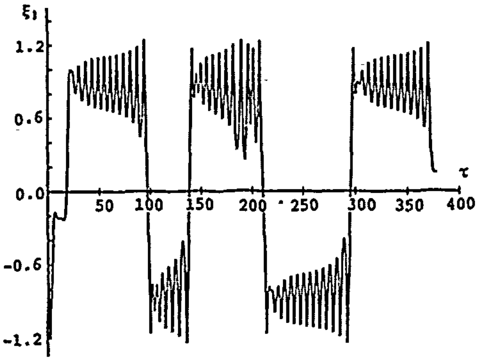


Fig.2. Time history for $\alpha_3=0.8$ (R=0.1).

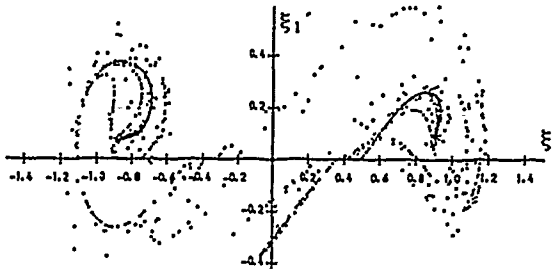


Fig.3. A strange chaotic attractor for $\alpha_3=0.3$

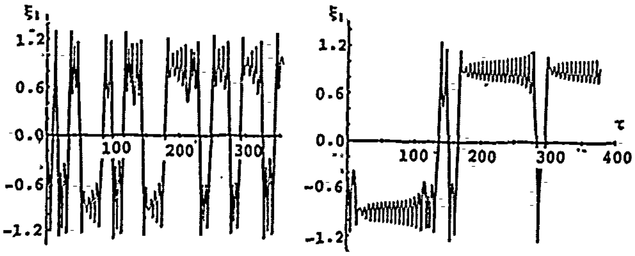


Fig.4. Two different time histories from the same chaotic attractor.

4. CONCLUSIONS

In the six-dimensional nonlinear mechanical system with friction that was investigated, quasiperiodic and chaotic attractors are detected. We have discussed and illustrated that in this case, the route to chaos is the same as in the simple two-well potential, sinusoidally-driven oscillator. An investigation of the influence of dry friction on the chaotic behavior of two coupled oscillators shows that increasing the friction weakens the chaotic dynamics of the orbits. During the chaotic motion of the first oscillator, stick-slip transitions are observed.

REFERENCES

1. J. Awrejcewicz, W.-D. Reinhardt, Some Comments About Quasi-Periodic Attractors, Journal of Sound and Vibration 139(2), 1990, 347.

2. J. Awrejcewicz, W.-D. Reinhardt, Quasiperiodicity, Strange Non-Chaotic and Chaotic attractors in the Forced System with Two Degrees of Freedom, Journal of Applied Mathematics and Physics ZAMP, 41, 1990, 713.

3. J. Awrejcewicz, W.-D. Reinhardt, Observation of Chaos in the Nonautonomous System with Two Degrees of Freedom, Journal of Applied Mathematics and Mechanics (in press).

"Indirect" Time Series Analysis for One-Dimensional Chaos Based on Perron-Frobenius Operator

Tohru KOHDA† and Kenji MURAO††

Kyushu University† and Miyazaki University††
Fukuoka 812† and Miyazaki 889-12††, Japan

Abstract A unified approach to time series analysis for one-dimensional discrete chaos is given which is based on the Galerkin approximation to the Perron-Frobenius integral operator. The proposed method gives approximations with high accuracy to statistics of various chaos. Numerical results for $1/f^{\delta}$ power spectrum of intermittent chaos also show that in the limit of zero frquencies, the observed exponent δ of the FFT power spectrum of long-time trajectories is in good agreement not with the Procaccia-Schuster's estimate but with ours.

I. Introduction

There are two kinds of time series analysis for long-time chaotic trajectories $\{x_m\}_{m=0}^{\infty}$ generated by a 1-d discrete dynamical system $x_{m+1} = \tau(x_m)$, $\tau : I = [0,1] \rightarrow I$. One of them is the "*time-average technique*", in which we evaluate certain statistics of a sample long-time trajectory $\{x_m\}_{m=0}^{n}$ with some initial value $x = x_0$; the other one is the "*ensemble-average technique*" under the assumption that τ is mixing with respect to an absolutely continuous invariant measure, denoted by $f^*(x)dx$. We give a unified approach to time series analysis for chaos by such an ensemble-average technique.

The time-average technique which is a usual method [2] is referred to as the "*direct method*". On the contrary, the ensemble average technique is a kind of "*indirect methods*" because there is no need to directly calculate trajectories. Hence such an indirect method is expected to play an important role in theoretically understanding chaos. In fact, the existence of $f^*(x)dx$ permits us to theoretically calculate the ensemble average of several statistics by using the Perron-Frobenius operator, denoted by P_τ [3],[4]. This operator, however, gives no practically calculating method because of its infinite dimensionality. Such a situation leads us to consider an efficient algorithm for systematically calculating statistics which is based on the Galerkin approximation to P_τ on a suitable function space[5]. Numerical experiments demonstrate that the proposed method can give approximations with high accuracy to statistics of various chaos.

II. Perron-Frobenius Operator and Statistics of Chaos

If $y = \tau(x)$ is mixing with respect to $f^*(x)dx$, then for almost initial value $x = x_0$ sequences $\{x_m\}_{m=0}^{\infty}$ can chaotically behave. From the Birchoff individual ergodic theorem, the time average of any L_1 function $F(x)$ along a trajectory $\{x_m\}_{m=0}^{\infty}$, which is defined by $\overline{F} = \lim_{T\rightarrow\infty} \frac{1}{T}\sum_{n=0}^{T-1} F(\tau^n(x))$, is equal almost everywhere to the ensemble average of $F(x)$ over I, defined by $< F >= \int_I F(x)f^*(x)dx$. The direct time series analysis is based on using $\overline{F}$. However, the *sensitive dependence on initial conditions*, one of chaotic properties[1], prevents us from precisely evaluating $\overline{F}$. On the other hand, the indirect time series analysis is based on using $< F >$. We begin with reviewing relations between typical statistics and P_τ.

The operator P_τ is defined by $P_\tau f(x) = \int_I \delta(x - \tau(y))h(y)dy$. For any L_1 functions of bounded variations $g(x)$ and $h(x)$, P_τ has the important property $(g(x), h(\tau(x))) = (P_\tau g(x), h(x))$, where $(g,h) = \int_I g(x)h(x)dx$. The invariant density $f^*(x)$ which plays a key role in our indirect method is the eigenfunction of P_τ belonging to the eigenvalue 1, that is, $P_\tau f^*(x) = f^*(x)$. The autocorrelation function is defined by $\rho(k) = < x\tau^k(x) > - < x >^2$. The first term of the rhs of this equation is rewritten as $< x\tau^k(x) >= (P_\tau^k x f^*(x), x)$, where the above property of P_τ is repeatedly used. Let $h_i(x)$ be the eigenfunction of P_τ with the eigenvalue λ_i for the eigenvalue problem $P_\tau h_i(x) = \lambda_i h_i(x)$ [2]. If we can expand $xf^*(x)$ as $xf^*(x) = \sum_{i=1}^{\infty} \eta_i h_i(x)$, then we have $\rho(k) = \sum_{i=2}^{\infty} u_i \lambda_i^k$, the Fourier Transform of which gives the power spectrum $S(\nu)$

$$S(\nu) = \sum_{i=2}^{\infty} u_i \frac{1-\lambda_i^2}{(1-\lambda_i z)(1-\lambda_i z^{-1})} \qquad (1)$$

where $\lambda_1 = 1$, $u_i = \eta_i(x, h_i)$ and $z = exp(j2\pi\nu)$ with $0 < \nu < 1$. Oono and Takahashi [3] [4] demonstrated that the Fredholm theory of P_τ plays an important role in discussions of the power spectrum. It is, however, difficult to find exact solutions of eigenvalues and eigenfunctions of P_τ, because P_τ has the infinite dimensionality. Such a situation led us to consider an efficient algorithm of the indirect method.

III. Galerkin Approximations to the Perron-Frobenius Operator

Let Δ be a function space which is spanned by a vector basis function $\vec{\ell}(x)$. Each component of $\vec{\ell}(x)$, denoted by $\ell_{nk}(x)$, is an appropriately chosen piecewise polynomial of at most D degree whose combination approximates to $f^*(x)$ by the Galerkin method [5] such as $f^*(x) \simeq \mathbf{f}^t\vec{\ell}(x)$, where the superscript t denotes the transpose of the vector $\mathbf{f}$. Using $\vec{\ell}(x)$, we get $< x\tau^k(x) >\simeq \mathbf{f}^t(P_\tau^k\vec{\ell}(x), x)$. Furthermore, using the Galerkin method with $\vec{\ell}(x)$ on Δ, we approximate to $P_\tau\vec{\ell}(x)$ such as $P_\tau\vec{\ell}(x) \simeq \widehat{P}_\tau^t\vec{\ell}(x)$ which leads us to readily obtain $< x\tau^k(x) >\simeq \mathbf{f}^t(\widehat{P}_\tau^t)^k(\vec{\ell}(x), x)$, where the $N(D+1)\times N(D+1)$ matrix $\widehat{P}_\tau$ is referred to as the *Galerkin-approximated matrix of the Perron-Frobenius operator* where N and D are integers to be given below. The explicit form of $\widehat{P}_\tau$ is given in [5]. Let $\mathbf{h}_i$ be the i−th right eigenvector of $\widehat{P}_\tau$ with the eigenvalue λ_i for the easily tractable eigenvalue problem $\widehat{P}_\tau\mathbf{h}_i = \lambda_i\mathbf{h}_i$.

The constructing method of Δ is as follows. We divide I into N subintervals $\{I_n\}$ with partition points $\{c_i\}_{i=0}^{N}$ satisfying $0 = c_0 < c_1 < c_2 < \quad < c_N = 1$ such that $I = \bigcup_{n=1}^{N} I_n$, $I_n = [c_{n-1}, c_n]$. The above Galerkin approximations depend on the appropriate selections of $\{c_i\}_{i=0}^{N}$ and of $\vec{\ell}(x)$ [5]. A simple but efficient procedure, however, is omitted here for selecting $\{c_i\}_{i=0}^{N}$. Next, we take bases $\ell_{nk}(x)$ such as $\ell_{nk}(x) = p_{nk}(x)s(x)\chi_n(x)$, $0 \leq k \leq D$, $1 \leq n \leq N$. In the above equation, $\chi_n(x)$ is the characteristic function of I_n and $p_{nk}(x)$ is the k−th order Legendre's polynomial which is orthogonal to each other on I_n. For most of practical usages, we use $D = 2$. When τ has a bounded invariant density, the function $s(x)$, referred to as a supplementary function, is taken to be 1. On the other hand, τ has an unbounded invariant density, $s(x)$ is chosen to be a singular function which approximates to singularities of the unbounded invariant density and the inner product (g,h) must be also replaced by the weighted inner product $(g,h)_w = \int_I g(x)h(x)w(x)dx$ with the weighting function $w(x) = s^{-2}(x)$.

IV. Numerical Examples

Example 1 Let

$$\tau(x) = \begin{cases} ax^z + (a+b-ab)/b & 0 \le x \le x_p = (1-1/b)^{1/z} \\ -b(x^z - 1) & x_p < x \le 1 \end{cases}$$

This map can generate periodic chaos for suitable parameters. Figures 1 and 2 show $f^*(x)$ and the power spectrum $\tilde{S}_T(\nu)$ for periodic chaos of period 6 which are calculated by our method. In this calculation, we take $\{\tau^n(0)\}_{n=1}^{30}$ as the partition points $\{c_i\}_{i=1}^{N-1}$ so that edges of the support of $f^*(x)$ will coincide with the partition points. In the calculation of $\tilde{S}_T(\nu)$, the finite discrete Fourier transform of $\{\rho(k)\}_{k=0}^{T-1}$ $(T = 1,024 \times 6)$ is used instead of using (1). On the other hand, $S_{T,m}(\nu)$ is obtained by averaging $m = 200$ discrete Fourier transforms of trajectories of length T. The spectrum $\tilde{S}_T(\nu)$ is in good agreement with $S_{T,m}(\nu)$ except for fluctuations in the latter.

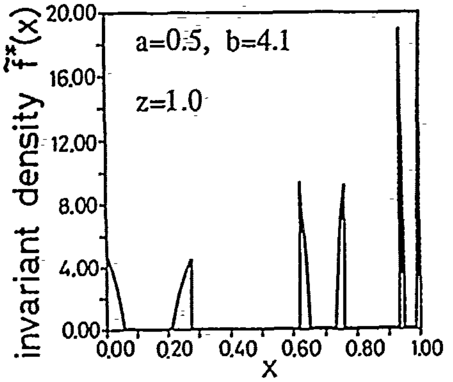


Fig. 1 Invariant density $f^*(x)$ (by our indirect method) for periodic chaos of period 6 in the example 1.

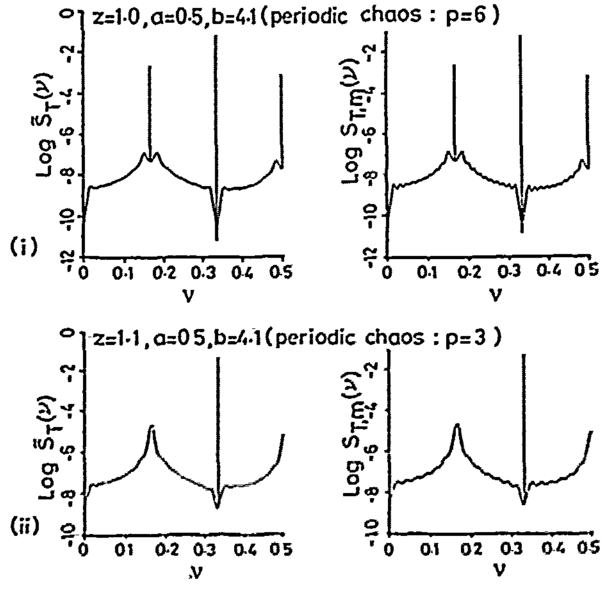


Fig. 2 Power spectra $\tilde{S}_T(\nu)$ (by our indirect method) and $\tilde{S}_{T,m}(\nu)$ (by the direct method) for periodic chaos of period 6 in the example 1.

Example 2 Let

$$\tau(x) = \begin{cases} x + ux^z & 0 \le x \le x_p \\ (x - x_p)/(1 - x_p) & x_p < x \le 1 \end{cases}$$

where $\tau(x_p) = 1, u > 0, 1 < z < 2$. This map generates intermittent chaos with the power spectrum $1/f^\delta$. Figure 3 shows the power spectrum $S(\nu)$ by our method (the smooth solid line) and $S_{T,m}(\nu)$ with $T = 2^{15}$ and $m = 100$ by the direct method (the fluctuated line), each of which is in good agreement each other in wide frequency range. In applying our method, we used $s(x) = x^{-(z-1)}$ because τ has the unbounded invariant density with a $(z-1)$-th order pole at $x = 0$. In this figure, the broken line shows the Procaccia and Schuster's estimate [6] of the spectrum when ν goes to 0 which does not coincide well with the former two.

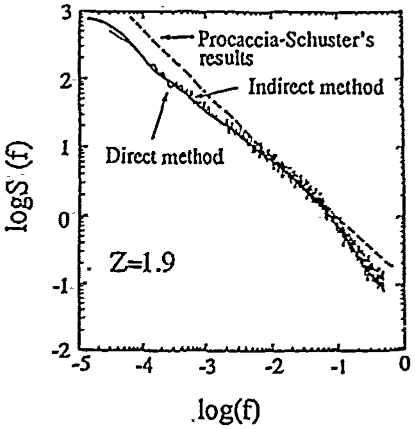


Fig. 3 Comparison of power spectra calculated by using three different methods for intermittent chaos in the example 2.

References

[1] S.Grossmann and S.Thomae, Z. Naturforsch., 32a, 1353-1363 (1977).

[2] S.J.Chang and J.Wright, Phys.Rev.A, 23, 1419-1433 (1981).

[3] Y.Oono and Y.Takahashi, Progr.Theo.Phys., 63-5, 1804-1807 (1980).

[4] Y.Takahashi and Y.Oono, Progr.Theo.Phys.. 71-4, 851-854 (1980).

[5] T.Kohda and K.Murao, Trans. IEICE, E73-6, 793-800 (1990).

[6] I.Procaccia and H.Schuster, Phys.Rev., A. 28-2, 1210-1212 (1983).

Constructive Implicit Function Theorem and its Application

Shin'ichi OISHI, Masahide KASHIWAGI, Mitsunori MAKINO and Kazuo HORIUCHI

School of Science and Engineering, Waseda University

Abstract

In this paper, a new algorithm of tracing solution curves for the homotopy method is presented. The algorithm is based on the predictor-corrector method and guaranteed that tracing solution curves always succeeds.

1.Introduction

Recently the study of the homotopy method have been made a great stride for solving nonlinear equations globally[1][2]. In the homotopy method, for solving a nonlinear equation

$$f(x)=0,\quad f:R^n\rightarrow R^n, \tag{1.1}$$

an auxiliary equation $g(x)=0$ is used, having a trivial solution $x_0\in R^n$. Changing the equation $g(x)=0$ into $f(x)=0$ gradually, a solution of $f(x)=0$ is obtained by tracing change of the solution of (1.1). For the purpose, we introduce a homotopy equation with parameter t

$$h(x,t)=0,\quad h:R^n\times[0,1]\rightarrow R^n, \tag{1.2}$$

where

$$h(x,0)=g(x),\quad h(x,1)=f(x). \tag{1.3}$$

Typically, the following homotopy h is used:

$$h(x,t)=(1-t)g(x)+tf(x). \tag{1.4}$$

Then a solution x^* of $f(x)=0$ can be obtained by tracing an implicitly defined solution curve $h^{-1}(0)$ from $(x_0,0)$ to $(x^*,1)$.

For tracing such a solution curve, a kind of predictor-corrector method is known to be effective. In this method, a point on the solution curve is moved along the tangent vector of the curve (predictor), and correct an deviation of the moved point from the solution curve $h^{-1}(0)$ (corrector) by, for example, the Newton method. However, the predictor-corrector method has a deficiency such that the method frequentry fails in tracing solution curves since the Newton iteration is used in the method. Although theoretically we can avoid the failure of tracing if we choose a step length sufficiently small, it has been said that to estimate such a step length is impossible[1][2].

In this paper, under the assumption that the derivative of h is Lipschitz continuous, we shall present "constructive" implicit function theorem. By the theorem, we can estimate the radius of a neighborhood in which implicit function theorem is valid. Moreover, the theorem gives a numerical algorithm to calculate the implicit function in that neighborhood based on the simplified Newton method.

Moreover, using the theorem, we present an improved predictor-corrector algorithm of tracing solution curves. By the new algorithm, tracing solution curves of (1.2) is guaranteed to succeed.

2.Constructive Implicit Function Theorem

In the following, when a Banach space X is dire[illegible]m of a subspace X_1 and a subspace X_2, we wro[illegible] X, when x is a sum of $x_1\in X_1$ and $x_2\in X_2$, as ([illegible]. Moreover, By $B(c;\varepsilon;X)$ we denote a ball with center c and radius ε in X.

The conventional implicit function theorem can be written as follows:

[Theorem 1](Implicit Function Theorem)

Let X be a Banach space which is direct sum of subspaces X_1 and X_2, Y a Banach space, $U\subset X$ an open set, and $g:U\rightarrow Y$ a C^1 operator. Assume that there exists a $(p_1,p_2)\in X$ such that the following conditions hold:

①$g(x_1,x_2)=0$,

②the partial derivative of g at (p_1,p_2) with respect to the second parameter is homeomorphism.

Then for sufficiently small $\varepsilon>0$ and $\delta>0$, the following holds true:

An equation

$$g(x_1,x_2)=0,\quad x_2\in B(p_2;\delta;X_2) \tag{2.1}$$

can be solved uniquely for x_2 with fixed $x\in B(p_1;\varepsilon;X_1)$.■

A map which maps x_1 to x_2 is called an implicit function. In this theorem, one can not estimate the largeness of ε and δ. To estimate them, we assume that the Frechet derivative of g, Dg, is α-Lipschitz continuous. Then the implicit function theorem can be extended as follows:

[Theorem 2](Constructive Implicit Function Theorem)

Let X be a Banach space which is direct sum of subspaces X_1 and X_2, Y a Banach space, $U\subset X$ an open set, and $g:U\rightarrow Y$ a CI operator. Assume that Frechet derivative of g, Dg, is α Lipschitz continuous. Moreover, we assume that a $(p_1,p_2)\in X$ and a bounded linear operator $A:X\rightarrow Y$ are given and that there exist r, K, d_1 and d_2 such that the following conditions hold:

① $\| g(p_1,p_2) \| \leqq r$, (2.2)

② $\| Dg(p_1,p_2) - A \| \leqq K$, (2.3)

③ $\| A(\cdot,0) \| \leqq d_1$, (2.4)

④There exists $A(0,\cdot)^{-1}: Y \rightarrow X2$ such that

$$\| A(0,\cdot)^{-1} \| \leqq 1/d_2. \quad (2.5)$$

Then if $\varepsilon > 0$ and $\delta > 0$ satisfy

$$\{\tfrac{1}{2}\alpha\varepsilon^2+(K+d_1)\varepsilon+r\}/\delta+\alpha(\varepsilon+\delta)+K \leqq d_2, \quad (2.6)$$

$$\alpha(\varepsilon+\delta)+K<d_2, \quad (2.7)$$

$$B(p_1;\varepsilon;X_1)\times B(p_2;\delta;X_2)\subset U, \quad (2.8)$$

the following holds true:

An equation

$$g(x_1,x_2)=0,\ x_2\in B(p_2;\delta;X_2) \quad (2.9)$$

can be solved uniquely for x_2 with fixed $x_1 \in B(p_1;\varepsilon;X_1)$ by the simplified Newton algorithm.■

Here we note that r and K represent the errors of (p_1,p_2) and A. If r=K=0, the situation is like that of theorem 1. To save a space, we omit the proof of theorem 2.

3.An Algorithm of Tracing Solution Curves

In this section, we consider to use theorem 2 to improve the predictor-corrector method. An outline of the idea is as follows. In what follows we restrict ourselves to a problem of tracing a solution curve of a equation h(x)=0, $h: R^{n+1} \rightarrow R^n$. Thus $X=R^{n+1}$, $Y=R^n$ and g=h in theorem 2. Let us consider a situation in which using a point x_i near a solution curve, a new point x_{i+1} on the solution curve is desired to be calculated. For the purpose, we calculate an approximation of $Dh(x_i)$ $A:X\rightarrow Y$ using for example numerical differentiation. Then we calculate the tangent vector of the solution curve and decide 1-dimensional subspace of X, X_1, containing the vector. Moreover we decide n dimensional subspace of X, X_2, to be the orthogonal complement of X_1. Then we calculate $\varepsilon>0$ and $\delta>0$ satisfying Eq.(2.6) and (2.7). If we choose a predictor q such that $\| q \| \leqq \varepsilon$, we can obtain a new point approximately on the solution curve by the following iteration in hyperplane $(p_1+q,\cdot)$:

$$y_0=p_2,\ y_{n+1}=y_n-A(0,\cdot)^{-1}h(p_1+q,y_n), \quad (3.1)$$

where $(p_1,p_2)=x_i$ and the new point x_{i+1} is given by (p_1+q,y_N), N is sufficiently large. Tracing solution curve is executed by repeating above mentioned process.

In the following we shall describe the algorithm more concretely. Let X be a (n+1)-dimensional Euclidian space E^{n+1}, introduced a Hilbert space by the canonical inner product. Then we determine X_1 and X_2 concretely as follows. Predictor $q \in X$ is decided to satisfy

$$Aq=0,\ q\neq 0, \quad (3.2)$$

where A is an approximate matrix of $Dh(x_i)$. Then X_1 is determined to be [q], a subspace which is generated by q, and X_2 is decided to be $[q]^{\mathsf{T}}$, an orthogonal complement of X_1. Here we note that in such decision, d_1 in Theorem 2 is estimated to be 0. Then $A(0,\cdot)^{-1}$ exists if A is maximal rank.

We now present the algorithm for tracing solution curves.

[Algorithm 1](Tracing Solution Curves)

Let X be a E^{n+1}, Y a E^n, $h:X\rightarrow Y$ a C^1 operator. Assume that Dh is α-Lipschitz continuous. Then we consider to trace solution curves of the equation h(x)=0. Let $x_0 \in X$ be a starting point on a solution curve, and $q_{-1} \in X$ a vector representing a prediction direction.

①Let i=0.

②Calculate $h(x_i)$ and $r=\| h(x_i) \|$.

③Calculate an approximate matrix of $Dh(x_i)$ A_i and an error estimation K such that $\| Dh(x_i)-A_i \| \leqq K$.

④Let q_i be a solution of the equation such that

$$q_i \in N(A_i),\ \| q_i \| =1,\ q_{i-1}\cdot q_i>0. \quad (3.6)$$

⑤Decide X_1 and X_2 to be [q] and $[q]^{\mathsf{T}}$.

⑥Calculate $\varepsilon>0$ and $\delta>0$ such that Eq.(2.6) and Eq.(2.7) is satisfied. Here let $d_1=0$ and $d_2=1/\| A_i(0,\cdot)^{-1} \|$.

⑦Execute the iteration such that

$$y_0=x_i+\varepsilon q_i,$$

$$y_{k+1}=y_k-A_i(0,\cdot)^{-1}(y_k), \quad (3.7)$$

until $\| h(y_k) \|$ becomes sufficiently small.

⑧Let $x_{i+1}=y_k$ and i=i+1. Go to ②. ■

4.Concluding Remarks

In this paper, assuming that the derivative of h is α-Lipschitz continuous, we have presented constructive implicit function theorem which can estimate the radius of the neighborhood in which implicit function theorem is valid. Using the theorem, based on the predictor-corrector method, we have presented a new algorithm of tracing solution curve without failure in the homotopy method.

This paper is partially supported by the Grant-in-Aid for the Ministry of Education, Science Research and Culture.

References

(1) C.B.Garcia and W.I.Zangwill: "Pathways to Solutions, Fixed Points and Equilibria " , Prentice-Hall(1981).

(2) E.L.Allgower and K.Georg: "Numerical Continuation Methods" , Springer-Verlag(1990).

A COMPARISON BETWEEN FINITE-DIFFERENCE, FINITE-ELEMENT, AND ALGEBRAIC MULTICONFIGURATION HARTREE FOCK APPROACHES FOR ATOMIC AND MOLECULAR CALCULATIONS

DAGE SUNDHOLM
Department of Chemistry, University of Helsinki
Et. Hesperiank. 4, SF-00100 Helsinki, Finland

AND

JEPPE OLSEN
Theoretical Chemistry, Chemical Centre, University of Lund
P.O. Box 124, S-22100 Lund, Sweden

Abstract: The finite-difference, finite-element, and the algebraic multiconfiguration Hartree-Fock methods are briefly compared, and some advantages and disadvantages of the approaches are discussed.

1. INTRODUCTION

In traditional quantum chemistry and physics, the molecular or atomic orbitals (one-electron functions) are expressed as linear combinations of global basis functions of the Slater ($\exp(\zeta r)$) or Gaussian ($\exp(-\zeta r^2)$) type. The truncation of the expansion leads to a basis-set truncation error (BSTE), the exact magnitude of which is difficult to establish. With recent advances in the treatment of the electron correlation, the BSTE has become a serious bottleneck in ab initio quantum chemistry. The BSTE can be systematically reduced to negligible magnitude by using numerical rather than algebraic approximation.

In the numerical approximation the unknown functions (orbitals) are expanded in piecewise differentiable local basis functions, while in the algebraic approximation global functions are used. In engineering sciences low-order polynomials can be used as local basis functions, while to achieve the high accuracy needed in quantum mechanical problems polynomials of fourth to eight order are used. The numerical approach can be divided into at least two different classes, the finite-difference (FD) and the finite-element (FE) methods.

In this paper, the algebraic, the finite-difference, and the finite-element approaches in quantum mechanical problems will be compared, and some of the advantages and disadvantages of these methods will be discussed. In order to concentrate on the differences between the three approaches, we will restrict the discussion to atomic systems. For molecules, there are numerical methods for solving various local-density functional (LDF) equations using FD method [1,2], FE method [3,4], and splines in 2D [5] and in 3D [6] based on one-centre expansions. Fully numerical approaches to Hartree Fock equations for diatomic molecules using FD method [2,7,8], FE method [9,10], and partial wave expansions [11,12] have been developed. Multiconfiguration Hartree Fock equations for diatomic molecules have also been solved by using FD method [2,13] and partial wave expansions [12,14].

2. THE MCHF METHOD

For atoms, the orbitals are expressed in spherical coordinates (r,θ,φ), as products of a radial and an angular part

$$\phi_i = R_i(r)\, Y_{l(i)}^{m(i)}(\theta,\varphi) \tag{1}$$

The angular parts Y_l^m are spherical harmonics, while the radial parts $R(r)$ are expanded in local or global basis functions. Using the second-quantization formalism with normalized wave functions, the multiconfiguration Hartree-Fock energy is given as

$$E = \sum_{i,j}^{occ} h_{ij}\, \Gamma_{ij} + \tfrac{1}{2} \sum_{i,j,k,l}^{occ} g_{ijkl}\, \Gamma_{ijkl} \tag{2}$$

where h_{ij} and g_{ijkl} are the one- and two electron integrals, and Γ_{ij} and Γ_{ijkl} are the elements of the one- and two electron density matrices, respectively.

$$h_{ij} = \int \phi_i^*(r) \left[-\tfrac{1}{2}\nabla^2 - Z/r \right] \phi_j(r)\, dr \tag{3}$$

$$g_{ijkl} = \int \phi_i^*(r_1) \left[\int \phi_k^*(r_2)\, (1/r_{12})\, \phi_l(r_2)\, dr_2 \right] \phi_j(r_1)\, dr_1 \tag{4}$$

$$\Gamma_{ij} = \langle 0 \mid \hat{E}_{ij} \mid 0 \rangle \tag{5}$$

$$\Gamma_{ijkl} = \langle 0 \mid \hat{E}_{ij}\, \hat{E}_{kl} - \delta_{jk}\, \hat{E}_{il} \mid 0 \rangle \tag{6}$$

where ϕ_i are the occupied orbitals, $-\tfrac{1}{2}\nabla^2$ is the kinetic energy operator, Z is the nuclear charge, r_{12} is the interelectronic distance, $\mid 0\rangle$ is the wave function, and $\hat{E}_{ij}$ are the excitation operators [15].

In the MCHF method both the expansion coefficients of the orbitals (occurring in h_{ij} and g_{ijkl}) and the coefficients of the configuration state functions (occurring in Γ_{ij} and Γ_{ijkl}) are optimized. In the Hartree Fock method the number of configurations is restricted to one, and the density matrices are thus fixed.

3. THE FD-MCHF APPROACH

In the finite-difference (FD) approach, the Euler (Fock) equations are obtained from the energy functional (2), and the Laplacian is discretized using n-point formulae. In engineering sciences low-order (n=2-3) polynomials are mostly used, while to achieve the high accuracy needed in quantum mechanical problems, one has to use high-order approximations (n=5 9/dimension). For each orbital i, the discretized coupled systems of non-linear Fock equations

$$\left[-\tfrac{1}{2}\nabla_i^2 - Z/r_i + 2 \sum_{j=1}^{occ} (V_{jj} - V_{ii} - \epsilon_{ii}) \right] \phi_i = \sum_{\substack{j=1 \\ j\neq i}}^{occ} (V_{ij} + \epsilon_{ij})\, \phi_j \tag{7}$$

are solved. The electron-electron interaction potentials, V_{ij}, are obtained using the Poisson equation

$$\nabla^2 V_{ij} = -4\pi\, \phi_i^* \phi_j \tag{8}$$

The Lagrange multipliers of equation (7), ϵ_{ij}, which ensure the orthonormality of the orbitals, are calculated as expectation values. The equations (7) and (8) are solved until the residual vanishes and the energy becomes stationary. For simplicity, we above assumed the Hartree-Fock approximation. In the MCHF method, the configuration interaction (CI) coefficients are also optimized. This is done by constructing the Hamilton matrix of the chosen configuration space and diagonalizing it. After the diagonalization, the potentials which now include the configuration interaction are recalculated, and the orbitals are reoptimized. This cycle is repeated until the changes of the orbitals and those of the CI coefficients are negligible. A more detailed description of the FD-MCHF method is given in refs. [16,17].

4. THE ALGEBRAIC MCHF (MCSCF) APPROACH

In the algebraic approach, the orbitals of the energy functional (2) are expanded in global Slater or Gaussian functions. The energy function obtained is optimized with repect to the orbital parameters and the CI coefficients, with imposed orthonormality constraints. The orbital and the CI parameters appear in the integrals and the density matrices, respectively. The main features of the MCSCF optimization will be described here. For details, the reader is referred to the literature [15,18].

The variation of the MC energy function is described by two unitary operators $\exp(\hat{S})$ and $\exp(\hat{T})$ for orbital and configuration parameter rotations, respectively. When the unitary operators are applied on a given state, the transformed wave function will remain normalized and the orbitals orthonormal. The $\hat{S}$ and $\hat{T}$ operators contain the independent variational parameters of the MC energy function. The Taylor series expansion of the energy function with respect to the parameters of the $\hat{S}$ and $\hat{T}$ can be constructed, and at the stationary point the first-order term (gradient) vanishes. The common choices for solving the optimization equations are the Newton-Raphson procedure or the Hessian update methods (quasi-Newton methods) [19,20]. The CI coefficients are usually calculated by using direct methods instead of explicit diagonalization of the Hamilton matrix.

5. THE FE-MCHF APPROACH

In the FE approach, the orbitals are expressed in local piecewise differentiable functions. We use Lagrange interpolation polynomials as basis functions [10]. Similarly to the algebraic approach an energy function can be constructed in the numerical basis. However, it is not possible to construct a fully orthonormal basis. The number of basis functions is huge. A basis where the occupied orbitals are orthonormal and orthogonal to those in the unoccupied space, and the unoccupied orbitals are non-orthonormal can be constructed. A transformation that rotates the virtual space into that of the occupied orbitals with out changing the orthonormality properties can also be made [22]. By using this generalized exponential mapping, it is possible also in the numerical case to parametrize the orbital rotations like in the algebraic approach. The exact gradient and the vector obtained by multiplying the Hessian matrix on the update vector can be calculated. The optimization equations can be solved using the Newton-Raphson or quasi-Newton methods. The configuration interaction parameters can be optimized as in the algebraic approach using the direct CI technique [20,21]. The numerical methods are discussed in more detail in refs. [22-24].

The FE-MCHF atomic structure package was used recently in the calculation of the electron affinity of boron [24]. In the largest calculation (1s inactive and 4 electrons in the 5s5p4d3f valence shulls), the number of configuration state functions (CSF) in $D_{\infty h}$ symmetry was 105447. The electron affinity obtained was 0.2668(30) eV as compared to the experimental value of 0.277(10) eV [25].

6. COMPARISON

The FE and algebraic methods are variational, while the FD method is not. In the FD method, the variational feature is used for deriving the Fock equations which then are discretized, while in the FE and algebraic methods the energy functional is first discretized (expanded in basis functions) and then varied. In general the FD matrix equations are unsymmetric, while the FE and the algebraic approaches result in symmetric matrix equations. The matrices of the FD method are more sparse than those of the FE method.

The FE and the algebraic matrix problems can also be seen as optimization problems, which easily can be controlled automatically. after each change of orbitals the energy should decrease, otherwise one has to go back to the previous point and try again. This is not true for FD method. In that case, the orbitals are adjusted until equations (7) and (8) are satisfied, and the energy may go up or down after each change of the orbitals. Second-order convergence methods (Newton-Raphson) can easily be used in the FE and algebraic methods, while it is not obvious how to implement them into the FD method.

In the numerical FD and FE methods, only integrals of the occupied space are needed. All integrals are recalculated after each change of orbitals. By doing this the storage and the computation of the two-electron integrals of the unoccupied space and the time consuming integral transformation ($\propto N^5$ operations, where N is the number of basis functions) are avoided.

The basis of the FD and FE methods cannot be fully orthonormalized because the number of basis functions is huge, while the equations in the algebraic approach are usually solved in an orthonormal basis. The main advantage of the FD and FE methods is the systematic convergence towards the limit of the model with an increasing number of basis functions.

In the FE and algebraic methods the error of the energy is quadratic in the error of the wave function, while in the FD method these errors are of the same order. Therefore it is incorrect to claim that the accuracy of the algebraic approach is comparable to or better than the accuracy of the FD method [26,27] even though the energy may have the same accuracy. The accuracy of a given property, the operator of which does not commute with the Hamilton operator, has the same accuracy as the energy in the FD approximation but not in the algebraic approach. We conclude that the algebraic approach cannot compete with the FE method as far as accuracy is concerned. The comparison is summarized in table 1.

Acknowledgement

The research reported in this article has been supported by The Academy of Finland, the Swedish Natural Science Research Council, and the Nordic Council of Ministers. This work has also been supported by IBM Sweden under a joint study contract.

Table 1. A comparison between FD, algebraic and FE methods

	FD	Algebraic	FE
Variational	No	Yes	Yes
Symmetric matrices	No	Yes	Yes
Sparce matrices	Yes	No	Yes
Automatic optimization control	No	Yes	Yes
Second-order convergence methods	No	Yes	Yes
Number of two-electron integrals [a]	n^4	N^4	n^4
Integral transformation	No	Yes	No
Well defined convergence with increasing size of the basis	Yes	No	Yes
Orthonormal basis	No	Yes	No

a) n is the number of occupied orbitals and N is the number of basis functions.

References:

1. L. Laaksonen, D. Sundholm, and P. Pyykkö, Intern. J. Quantum Chem. 27 (1985) 601.
2. L. Laaksonen, P. Pyykkö, and D. Sundholm, Comp. Phys. Rep. 4 (1986) 313; and references therein.
3. D. Heinemann, B. Fricke, and D. Kolb, Phys. Rev. A38 (1988) 4994.
4. D. Heinemann, A. Rosén, and B. Fricke, Chem. Phys. Letters 166 (1990) 627.
5. A.D. Becke, J. Chem. Phys. 76 (1982) 6037.
6. A.D. Becke, J. Chem. Phys. 88 (1988) 2547.
7. L. Laaksonen, P. Pyykkö, and D. Sundholm, Chem. Phys. Letters 96 (1983) 1.
8. K. Davstad, Ph.D Thesis, University of Stockholm, Sweden (1990).
9. D. Heinemann, D. Kolb, and B. Fricke, Chem. Phys. Letters 137 (1987) 180.
10. D. Sundholm, J. Olsen, P.Å. Malmqvist, and B.O. Roos, in "Numerical Determination of the Electronic Structure of Atoms, Diatomic and Polyatomic Molecules", eds. M. Defranceschi and J. Delhalle, (Kluwer Dordrecht, 1989) p. 329.
11. E.A. McCullough Jr., Chem. Phys. Letters 24 (1974) 55.
12. E.A. McCullough Jr. Comp. Phys. Rep. 4 (1986) 265; and references therein.
13. L. Laaksonen, D. Sundholm, and P. Pyykkö, Chem. Phys. Letters 105 (1984) 573.
14. E.A. McCullough Jr., J. Phys. Chem. 86 (1982) 2178.
15. Three recent review articles by B.O. Roos, R. Shepard, and H.J. Werner can be found in: "Ab initio Methods in Quantum Chemistry Part II; Adv. Chem. Phys. 69; ed K.P. Lawley, (Wiley, Chichester, U.K. 1987).
16. C. Froese Fischer, "The Hartree-Fock Method for Atoms", (Wiley, New York, 1977).
17. C. Froese Fischer, Comp. Phys. Rep 3 (1986) 273.
18. G.H.F. Diercksen and S. Wilson (eds.), "Methods in Computational Physics", (Reidel, Dordrecht, 1983).
19. R. Fletcher, "Practical Methods of Optimization", (Wiley, New York, 1980) Vol 1.
20. J. Olsen, D.L. Yeager, and P. Jørgensen, Adv. Chem. Phys. 54 (1983) 1.
21. J. Olsen, P. Jørgensen, and J. Simons, Chem. Phys. Letters 169 (1990) 463.
22. J. Olsen and D. Sundholm, (to be published).
23. D. Sundholm and J. Olsen, Phys. Rev. A42 (1990) 1160; Phys. Rev. A42 (1990) 2614; Chem. Phys. Letters 177 (1991) 91.
24. D. Sundholm and J. Olsen, Chem. Phys. Letters 171 (1990) 53.
25. H. Hotop and W.C. Lineberger, J. Phys. Chem. Ref. Data 14 (1985) 731.
26. B.H. Wells and S. Wilson, J. Phys. B: At. Mol. Opt. Phys. 22 (1989) 1285.
27. J.W. Thompson and S. Wilson, J. Phys. B: At. Mol. Opt. Phys. 23 (1990) 2295.

AN ALGORITHM FOR THE LOCATION OF FIRST-ORDER SADDLE-POINT

P Culot, G Dive
Centre d'Ingénierie des Protéines
Université de Liège
Institut de Chimie, B6
B-4000 Liège (Sart-Tilman)
Belgium

VH Nguyen
Département de Mathématiques
Facultés Universitaires de Namur
Rempart de la Vierge, 8
B-5000 Namur
Belgium

I. INTRODUCTION

According to Eyring and Polanyi [1], a chemical reaction coordinate can be seen as a path going from an energy minimum reactant state via a transition state to an energy minimum product state. The transition state is characterized by a maximum energy along the reaction path. This stationary point is maximum along the reaction coordinate and minimum in all other orthogonal directions. Thus, this first-order saddle-point on the potential energy surface [2] is associated to an indefinite Hessian matrix with only one negative eigenvalue.

The purpose of this paper is to propose an augmented quasi-Newton algorithm to locate a first-order saddle-point. The algorithm is compared to the efficient method of Baker [3] in the study of the methanolysis of protonated methyl-formic-ester.

II. ALGORITHM

A. Quasi-Newton step

The energy E(x), a function of n real variables, is, at least, twice continuously differentiable. A quadratic approximation of the energy function around the current point x can be considered as,

$$E(x+D) \approx Q(D) = E + G^tD + \frac{1}{2}D^tHD \qquad (1)$$

where E, G and H are the energy function, the gradient vector and the Hessian matrix evaluated at the current point x, respectively. D is a displacement vector around the current point.

The stationary point of Q(D) is the quasi-Newton step :

$$D = -H^{-1}G. \qquad (2)$$

This step can be written as,

$$D = -\sum_{i=1}^{n} \frac{\overline{G}_i}{b_i} V_i$$

where b_i and V_i are the eigenvalues and eigenvectors of the Hessian H. $\overline{G}_i$ is the component of the gradient vector along the eigenvector V_i.

If the Hessian matrix H is indefinite with one negative eigenvalue, then the quasi-Newton step is a good search direction. But, if the Hessian does not have this expected inertia, then the Hessian matrix has to be perturbed in order to obtain a new step calibration which is ascendent in one direction and descendent along all the orthogonal ones. The new proposed algorithm gives rise to an augmented quasi-Newton step.

B. Augmented quasi-Newton step

The quadratic approximation Q(D) is only significant near the current point. A scaling of the direction step is then done via a restricted step method. The displacement vector is chosen inside a trust region,

$$\Omega = \{D : D^tD \leq R^2\} \qquad (4)$$

with a trust radius R. Inside this trust region, the transition structure search step is calculated via a maximisation of the quadratic approximation along an eigenvector V_1 and a minimisation of the approximation along the other eigenvectors.

The solution of this optimization problem generates the augmented quasi-Newton step

$$D(\lambda) = -\frac{\overline{G}_1}{b_1 - \lambda} V_1 - \sum_{i=2}^{n} \frac{\overline{G}_i}{b_i + \lambda} V_i \qquad (5)$$

The positive parameter λ is chosen such that

(i) the search direction is inside the trust region Ω,
(ii) the augmented Hessian matrix is indefinite with one negative eigenvalue. Thus, the conditions

$$b_1 - \lambda < 0 \text{ and } b_i + \lambda > 0, i=2,\ldots,n \qquad (6)$$

have to be fulfilled.

If the Hessian matrix has the expected inertia and if the quasi-Newton step is inside the trust region, then the quasi-Newton step is selected. Otherwise, the step is chosen on the boundary of the trust region with a parameter λ solution of

$$D^t(\lambda)D(\lambda) = R^2 \qquad (7)$$

$$\lambda \geq 0 \qquad (8)$$

$$\lambda > \max\{b_1, -b_2\} \qquad (9)$$

III. APPLICATION

The algorithm is used for the location of a transition state arrangement of the methanolysis of protonated methyl-formic-ester (figure 1). The results are compared to those derived from the method of Baker [3].

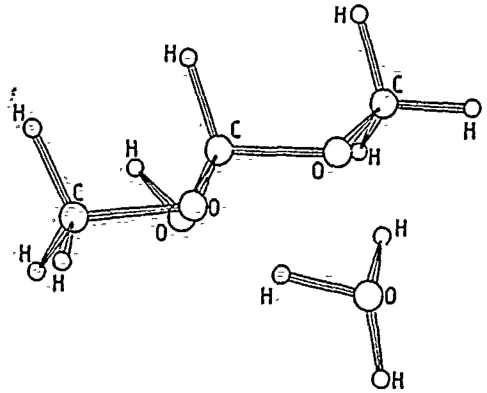


Figure 1. Methanolysis of protonated methyl-formic-ester. First-order saddle-point located by the augmented quasi-Newton algorithm.

The energy surface associated to this protonated system has a low curvature. From a starting point, which is not taken within the quadratic region of the solution, the algorithm converges to the saddle point (figure 1) after 20 iterations. The eigenvector components associated to the -0.00139 eigenvalue well explain the flip-flop of the water molecule from the hydrated methanol to the methoxy ester group. After 116 iterations, the algorithm of Baker [3] converges to a saddle point in which only the torsional angles of the methyl rotation are concerned. Therefore, this critical point looks like a complex between ester and the couple methanol-water (figure 2).

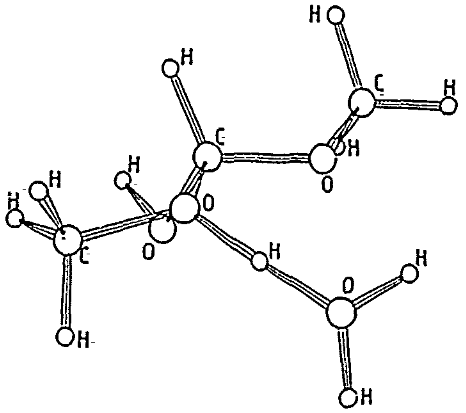


Figure 2. Methanolysis of protonated methyl-formic-ester. First-order saddle-point located by the algorithm of Baker.

Forcing the algorithm to retain the right Hessian inertia, the calculation of D without any guide, as given by the trust region method, can converge to saddle-point structures which are not related to the expected chemical rearrangement

IV. CONCLUSION

This paper deals with an algorithm involved in a transition state arrangement location on an energy surface. This algorithm solves the problem of step estimation as an augmented quasi-Newton displacement by adding a positive shift parameter. The efficiency is well illustrated by a 18-atom system associated with a very low curvature surface.

One important problem inherent to the matrix inertia requires further more investigations. Based on the Hessian inertia, the D step calculation involves a second derivative update which, at present time, has to be improved.

REFERENCE

[1] H Eyring, M Polalyi, Z Phy. Chem B12, 277 (1931)

[2] JN Murrel, KJ Laidler, Trans faraday Soc 64, 371 (1968)

[3] J Baker, J Comp Chem 7, 385 (1986)

MOMENTUM SPACE QUANTUM CHEMISTRY CALCULATIONS. PROBLEMS AND PROMISES.

MIREILLE DEFRANCESCHI

DSM-DRECAM-SPAS, CEN-Saclay,
F-91191 Gif-sur-Yvette Cedex (France)

and

JOSEPH DELHALLE

Laboratoire de Chimie Théorique Appliquée, Facultés
Universitaires N.D. de la Paix, 61, rue de Bruxelles
B-5000 Namur (Belgium)

The model of a molecule in which the nuclei and electrons are assumed to be non-relativistic point charges interacting through electrostatic (Coulomb) forces has been found to provide a satisfactory description of molecular properties [1]. *Ab initio* calculations in molecular quantum chemistry most often mean solving the time-independent Schrödinger equation $H\Psi_i = E\Psi_i$ where H is the Hamiltonian of a molecule based on that model, Ψ_i is the i-th wavefunction and E_i the corresponding energy eigenvalue. Even when complexity is reduced by considering only the motion of the electrons in a fixed nuclear framework (Born-Oppenheimer approximation), the inherent mathematical difficulties due to the multicenter nature of the electrostatic interactions are such that solutions are not obtainable in explicit form.

Finding suitable but manageable approximate solutions to the electronic Schrödinger equation has thus been a major preoccupation of quantum chemists. Central to attempts at solving such problems is the Hartree-Fock (HF) theory [1]. The essence of this approximation is to replace a complicated many-electron case by a one-electron problem in which the electron-electron interaction is treated in an average way.

Position and momentum HF equations.

Restricted to closed-shell systems with n electrons, the n/2 doubly occupied HF orbitals φ_i are obtained in position space as solutions of an integro-differential equation, $(F - \varepsilon_i)\,\varphi_i = 0$, where the HF operator F is a one-electron Hamiltonian. It includes a kinetic term and an effective potential itself comprising the electron-nucleus attraction and a Coulombic potential approximating the real electron-electron interactions. In atomic units, the equation writes as:

$$F\varphi_i(r) = -\frac{\Delta(r)}{2}\varphi_i(r) - \sum_u \frac{Z_u}{|r-R_u|}\varphi_i(r) + \int dr' \sum_j^{n/2} \frac{2\varphi_j^*(r')\varphi_j(r')}{|r-r'|}\varphi_i(r)$$

$$-\int dr' \sum_j^{n/2} \frac{\varphi_j^*(r')\varphi_i(r')}{|r-r'|}\varphi_j(r). \qquad (1)$$

Explicit solutions to eq(1) cannot be obtained because of the terms $|r-R_u|^{-1}$ and $|r-r'|^{-1}$. In position space, numerical solutions can be constructed for diatomic molecules, but not for polyatomic systems. In such cases approximate solutions are expressed as truncated linear combinations of basis functions (LCAO expansion). In spite of its successes, the LCAO approximation experiences various difficulties (truncation limits, nature of the basis functions, etc.) which are not entirely controllable [2]. Formulated in momentum space, the HF equations give way to numerical approaches in which Coulombic interactions become tractable even for polyatomic molecules [3] In momentum space, eq(1) becomes,

$$\left(\frac{p^2}{2}-\varepsilon_i\right)\phi_i(p) - \frac{1}{2\pi^2}\int\frac{dq}{q^2}\Big[\Big(S(q) - \sum_j^{n/2} 2W_{jj}^*(q)\Big)\phi_i(p-q)$$

$$-\sum_j^{n/2} W_{ij}^*(q)\,\phi_j(p-q)\Big] = 0, \qquad (2)$$

where the molecular structure factor S(q) and the interaction terms $W_{ij}(q)$ are:

$$S(q) = \sum_u Z_u \exp(iq.R_u) \qquad (3a)$$

$$W_{ij}(q) = \int dr\, \varphi_i^*(r)\varphi_j(r)\, e^{-iqr} = \int dp\, \phi_i^*(p)\,\phi_j(p-q). \qquad (3b)$$

Among other advantages, these equations do not require coordinate systems adapted to the geometry of the molecules to remove Coulombic singularities which make the the position space formulation numerically untractable beyond diatomic systems. In eq(2) the only singular contribution comes from the q^{-2} factor.

Numerical Procedure and Problems.

In both position and momentum spaces, iterative procedures are necessary to solve the HF equations. Starting from a trial orbital $\phi_i^{(0)}(p)$, an approximate orbital $\phi_i^{(k+1)}(p)$ is constructed after k+1 iterations of eq(2) rewritten as [4]:

$$\phi_i^{(k+1)}(p) = \Big[\frac{p^2}{2}-\varepsilon_i^{(k)}\Big]^{-1}\frac{1}{2\pi^2}\int\frac{dq}{q^2}\Big\{\Big(S(q) - 2\sum_j^{n/2} W_{jj}^{(k)*}(q)\Big)\phi_i^{(k)}(p-q)$$

$$+\sum_j^{n/2} W_{ij}^{(k)*}(q)\,\phi_j^{(k)}(p-q)\Big\}. \qquad (4)$$

Numerical and computational problems associated with the implementation of the approach for routine use fall in two main categories : (a) numerical integration, and (b) control of the orthogonality of the numerical orbitals during the iteration steps

Different integration schemes have been considered To advantageously cancel the singular q^{-2} factor in eq(4) by the integration volume element, Navaza and Tsoucaris [3] have proposed the use of spherical polar coordinates. However, because of the convolution integrals, interpolation schemes are needed in these coordinates since arguments (p - q) do not necessarily belong to the grid points. Another point of view has been to focus on these convolution integrals and treat them via a more economical fast Fourier transform procedure, but at the expense of an approximate treatment of the q^{-2} singular factor [5,6]. Variants [7,8] based on the Fock transformation [10] have also been proposed to deal with the infinite limits of integration. Computational tests [10] in the case of the helium atom have shown the importance of accuracy and convergence of the integrals and, at present, none of the approaches so far attempted has been satisfactory enough to bring the momentum quantum chemistry calculations beyond a stage of prematurity.

Orthonormalization also raises problems. At each step, the new iterates $\phi_i^{(k+1)}(p)$'s need to be renormalized and orthogonalized to form true canonical HF orbitals [1]. Great care must be exercised in selecting orthogonalization procedures, for instance the so-called Löwdin's symmetric orthogonalization procedure [11], pervasively used in quantum chemistry, mixes all the orbitals simultaneously, tends to contaminate the iterates, and impairs the convergence of the iterative steps [12]. Schmidt orthogonalization does better but looses track of the symmetry of these orbitals. Reformulation of eq(4) in a form with symmetry and orthogonality constraints would be very valuable.

Promises of the Approach.

In spite of the above problems, increasing number of results have been harvested with momentum space quantum chemistry calculations.

Fully numerical HF orbitals for a triatomic molecule have been obtained for the first time [13] with a procedure similar to that originally proposed by Navaza and Tsoucaris [3]. The qualitative and quantitative advantages of using high quality numerical HF orbitals to go beyong the HF level have also been pointed out [14].

With trial orbitals $\phi_i^{(0)}(p)$ expressed as linear combinations of gaussian functions, it is possible to work out the first iteration and write the first iterates $\phi_i^{(1)}(p)$ in terms of transcendental functions (e.g. Dawson function); the only numerical steps left being the normalization and orthogonalization. An analysis [15] carried out on the first iterates reveals that, by the virtue of its integral form, eq(4) imparts the right asymptotic behaviour to the first iterates $\phi_i^{(1)}(p)$ at large and small values of $|p|$. Recent investigations [12,16-18] on atomic systems of increasing complexity (H, He, H^-, Be, B^+, etc.) indeed show significant improvements, qualitative and quantitative, in energy and wavefunction properties. Thus, already with a first step, it is possible to correct for the deficiencies of orbitals expressed as truncated linear combination of basis functions and produced by standard quantum chemistry packages.

Acknowledgments. The authors thank ATOCHEM, CISI, the CEA and the Belgian National Fund for Scientific Research (FNRS) for financial support.

References.

1. R. McWeeny, *Methods of Molecular Quantum Mechanics,* (Academic Press, New York, 1989), 2nd ed.
2. G. Fonte, Theoret. Chim. Acta 59, (1981) 533.
3. J. Navaza, G. Tsoucaris, Phys. Rev. A24 (1981) 683.
4. N.V. Svartholm, Ark. Mat. Astron. Fys., 35A, n° 7 & 8 (1947).
5. S.A. Alexander, H.J. Monkhorst, Intern. J. Quantum Chem. 32 (1987) 361.
6 S A Alexander, R L Coldwell, H.J. Monkhorst, J. Comput. Phys. 76 (1988) 263.
7. W. Rodriguez, Y. Ishikawa, Chem. Phys. Lett. 146, (1988) 515.
8. Y. Ishikawa, I.L. Aponte-Avellanet, S.A. Alexander, Int. J. Quantum Chem. Symp. 23 (1989) 209.
9. V. Fock, Z. Phys. 98 (1935) 145.
10. J. Delhalle, M. Defranceschi, Int. J. Quantum Chem. Symp. 21 (1987) 425.
11. P.O. Löwdin, J. Chem. Phys 18 (1950) 365.
12. L. Dewindt, J.G. Fripiat, J. Delhalle, M. Defranceschi, J. Mol. Struct. (Theochem), in press.
13 M Defranceschi, M. Suard, G. Berthier, Comptes Rendus Acad. Sci. 296 (1983) 1301.
14. M. Defranceschi, M. Suard, G. Berthier, Comptes Rendus Acad. Sci. 301 (1985) 1405.
15. J. Delhalle, J.G. Fripiat, M. Defranceschi, Annales Soc. Scient. Brux., 101 (1987) 9.
16 M Defranceschi, J Delhalle, Eur. J. Phys., 11 (1990) 172.
17. J.G. Fripiat, J. Delhalle, M. Defranceschi, in *Numerical Determination of the Electronic Structure of Atoms, Diatomic and Polyatomic Molecules,* M. Defranceschi, J. Delhalle (eds.)NATO-ASI, vol. C271, (Kluwer Academic Publishers, Dordrecht, 1989), pp. 263-268.
18. J. Delhalle, J.G. Fripiat, M. Defranceschi, Bull. Soc. Chim. Belg. 99 (1990) 135.

DETERMINATION FROM EXPERIMENTAL MEASUREMENTS OF TRANSPORT COEFFICIENTS AT THE DIFFUSION TIME SCALE IN A TOKAMAK

J. BLUM[1], H. CAPES[2], Y. STEPHAN[3]

[1]Université de Grenoble, BP53X, 38041 Grenoble Cedex, France.
[2]Commissariat à l'Energie Atomique, C.E.N. Cadarache, 13108 Saint Paul lez Durance.
[3]CISI INGENIERIE, C.E.N. Cadarache, 13108 Saint Paul lez Durance, France.

Abstract : the knowledge and understanding of the particle and energy transport in a tokamak are of crucial interest to obtain controlled fusion. In fact, the confinement and stability of the plasma cannot be ensured without a whole comprehension of these phenomena. Our aim is then to determine numerically the transport coefficients which govern the equations of conservation from extraneous informations contained in experimental data. This is achieved by a 1D1/2 representation of the resistive MHD equations added to a least square formulation of the constraints. This optimal control problem is then solved using the linear quadratic sequential method and finite elements for space discretization.

1. THE EQUILIBRIUM AND TRANSPORT MODEL

At the diffusion time scale, the momentum conservation equation reduces and the equilibrium assumption holds at each time (cf. Ref [1] for a complete bibliography). In an axisymmetric configuration, it leads to the following 2D Grad-Shafranov equation :

$$L\Psi = j_t(\Psi) \qquad (1)$$

where :

Ψ is the poloidal flux of the magnetic field B,

$L = -\frac{\partial}{\partial r}(\frac{1}{\mu_0 r}\frac{\partial}{\partial r}) - \frac{\partial}{\partial z}(\frac{1}{\mu_0 r}\frac{\partial}{\partial z})$ is an elliptic operator,

j_t is the toroidal plasma current density, and

μ_0 is the magnetic permeability in air.

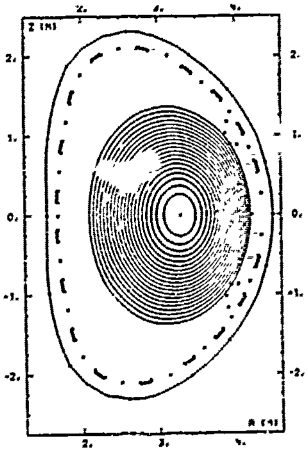


Fig.1 Flux lines obtained from IDENTD for an equilibrium at JET tokamak.

The determination of j_t from experimental data and the numerical resolution of equation (1) with appropriate boundary condition is operated by the software IDENTD (cf. Ref [1][2][3]). This gives the location of the nested magnetic surfaces where Ψ is a constant as shown in Fig.1.

The averaging of the conservation equations on each magnetic surface leads to the following 1D system (cf. Ref [1]):

Conservation of electrons :

$$\frac{\partial}{\partial t}(V' n_e) + \frac{\partial}{\partial \rho}(V'\Gamma_e) = V'\langle S_1\rangle \qquad (2)$$

where :

$\rho = (\frac{\phi}{\pi B_0})^{1/2}$ labels the magnetic surfaces,

ϕ is the toroidal flux of the magnetic field B,
B_0 is the magnetic field at a fixed point $r = R_0$,
n_e is the electronic density,

$V' = \frac{\partial V}{\partial \rho}$ where V is the volume enclosed by the magnetic surface ρ,
and $\langle S_1\rangle$ is a source term.
In the "diagonal" model, the particle flux is

$$\Gamma_e = -D\langle\nabla^2\rho\rangle\frac{\partial n_e}{\partial \rho}$$

Conservation of energy for electrons :

$$\frac{3}{2}\frac{1}{V'^{2/3}}\frac{\partial}{\partial t}(V'^{5/3}P_e) + \frac{\partial}{\partial \rho}(V'(Q_e + \frac{5}{2}kT_e\Gamma_e))$$
$$= V'(-\frac{\Gamma_e \partial P_e}{n_e \partial \rho} - \alpha(P_e-P_i) + S_{ohm} + \langle S_2\rangle) \qquad (3)$$

where :
$P_e = n_e kT_e$ is the electronic pressure,
T_e is the electronic temperature,
k is the Boltzmann constant,

$Q_e = -K_e\langle\nabla^2\rho\rangle\frac{\partial T_e}{\partial \rho}$ is the "diagonal" heat flux,

$\alpha(P_e-P_i)$ is the equipartition term,
P_i is the ionic pressure, S_{ohm} is a source term due to the Joule effect,

$$S_{ohm} = \frac{\eta\rho}{\mu_0^2 V' C_3^2}\frac{\partial}{\partial \rho}(C_2\Psi')\frac{\partial}{\partial \rho}(\frac{C_2 C_3 \Psi'}{\rho}),$$

$\Psi' = \frac{\partial \Psi}{\partial \rho}$, η is the resistivity, C_2 and C_3 are geometric coefficients,
and $\langle S_2\rangle$ is a source term.

Conservation of energy for ions :

$$\frac{3}{2}\frac{1}{V'^{2/3}}\frac{\partial}{\partial t}(V'^{5/3}P_i) + \frac{\partial}{\partial \rho}(V'(Q_i + \frac{5}{2}kT_i\Gamma_i))$$
$$= V'(\frac{\Gamma_i \partial P_i}{n_i \partial \rho} + \alpha(P_c-P_i) + \langle S_3\rangle) \qquad (4)$$

where :
$P_i = n_i kT_i$ is the ionic pressure,
T_i is the ionic temperature, n_i is the ionic density;

for reason of neutrality, $n_c = Zn_i$ and $\Gamma_c = Z\Gamma_i$, where Z is the mean charge of ions;

$Q_i = -K_i\langle\nabla^2\rho\rangle\frac{\partial T_i}{\partial \rho}$ is the "diagonal" heat flux for ions and $\langle S_3\rangle$ is a source term.

To determine the resistivity η, we use the resistive diffusion equation for the flux derivative,

$$\frac{\partial \Psi'}{\partial t} + \frac{\partial}{\partial \rho}(\frac{\eta\rho}{\mu_0 C_3^2}\frac{\partial}{\partial \rho}(\frac{C_2 C_3 \Psi'}{\rho})) = 0 \qquad (5)$$

and the averaged Grad-Shafranov equation :

$$\frac{\partial}{\partial \rho}(C_2\Psi') = -\mu_0 V'\langle\frac{j_t}{r}\rangle \qquad (6)$$

Boundary conditions : (7)

$$\frac{\partial n_e}{\partial \rho}(0,t) = \frac{\partial P_e}{\partial \rho}(0,t) = \frac{\partial P_i}{\partial \rho}(0,t) = \Psi'(0,t) = 0$$

$n_e(\rho_{max},t)$, $P_e(\rho_{max},t)$, $P_1(\rho_{max},t)$ are given, and either the total current $I_p(t)$ or the tension by lap $V(t)$ are known :

$C_2\Psi'(\rho_{max},t) = - 2\pi\mu_0 I_p(t)$ from (6); or

$$\frac{\eta\rho}{\mu_0 C_3{}^2}\frac{\partial}{\partial\rho}\left(\frac{C_2C_3\Psi'}{\rho}\right)(\rho_{max},t) = V(t)$$

2. THE INVERSE PROBLEM

At each time, we can calculate the equilibrium and determine the flux lines by the software IDENTD. The averaging technique gives the geometry $\rho(\Psi)$, $V'(\rho)$, $C_2(\rho)$, $C_3(\rho)$ of the magnetic surfaces, the profiles $n_e{}^m(\rho)$, $T_e{}^m(\rho)$, $T_i{}^m(\rho)$, obtained from experimental data by Abel inversion on the geometry of the flux lines, the current density $<\frac{jt}{r}>$ as a result of the identification procedure used in IDENTD, and the profile $\Psi'(\rho)$ by solving equation (6) with the boundary condition $\Psi'(0,t) = 0$. The idea is to solve the equilibrium problem at times t_1 and t_2 and then try to determine the transport coefficients D, K_e, K_i and η for which the equations (2) to (7) lead from the initial state at time t_1 to the final state at time t_2.

Setting $u = [D,K_e,K_i,\eta]$ and $y = [n_e,P_e,P_i,\Psi']$, the problem (P) is then to minimize over (u,y) solutions of equations (2) to (7) the following cost function :

$$\begin{aligned} J(u,y) = & \frac{K_{ne}}{2}\left|n_e(.,t_2)-n_e{}^m(.,t_2)\right|^2{}_V \\ & +\frac{K_{Pe}}{2}\left|T_e(.,t_2)-T_e{}^m(.,t_2)\right|^2{}_V \\ & +\frac{K_{Pi}}{2}\left|T_i(.,t_2)-T_i{}^m(.,t_2)\right|^2{}_V \\ & +\frac{K_{\Psi'}}{2}\left|\Psi'(.,t_2)-\Psi'^m(.,t_2)\right|^2{}_V \\ & +\frac{\varepsilon_D}{2}\left|\frac{\partial^2 D}{\partial\rho^2}\right|^2{}_V+\frac{\varepsilon_{Ke}}{2}\left|\frac{\partial^2 K_e}{\partial\rho^2}\right|^2{}_V \\ & +\frac{\varepsilon_{Ki}}{2}\left|\frac{\partial^2 K_i}{\partial\rho^2}\right|^2{}_V+\frac{\varepsilon_\eta}{2}\left|\frac{\partial^2 \eta}{\partial\rho^2}\right|^2{}_V \end{aligned} \quad (8)$$

where the last four terms are due to a Tichonov regularization technique, the K's and ε's are weighting factors, and $V = L^2(]0,\rho_{max}[)$.

3. NUMERICAL METHODS

The problem (P) is an optimal control problem equivalent to the determination of the saddle point of the following lagrangian (cf. Ref [4]) :

$\mathcal{L}(u,y,p) = J(u,y) + <F(u,y),p>$ where $F(u,y) = 0$ represents the set of the state equations (2) to (5), the adjoint state p their Lagrange multiplier, and $<q,p>$ the scalar product in $L^2(]t_1,t_2[x]0,\rho_{max}[)$. The optimality conditions for (P) give :

$$\frac{\partial\mathcal{L}}{\partial y}z = 0 \text{ for all } z, \quad (9)$$

or $<\frac{\partial F^*}{\partial y}p, z> = -\frac{\partial J}{\partial y}z$ which determines p, and the gradient

$$\frac{dJ}{du} = \frac{\partial\mathcal{L}}{\partial u} = \frac{\partial J}{\partial u} + <\frac{\partial F}{\partial u},p> \quad (10)$$

After a linearization with respect to u and y of F, in the same way the cost function J becomes quadratic, and we finally derive a discrete formulation using the finite element method. The problem is then solved by a conjugate gradient algorithm at each iteration of the Newton procedure.

4. NUMERICAL RESULTS

The validity of our tool has been checked by the following procedure : for given D, K_e, K_i, η and initial state, we first integrate the transport equations and then try to reconstruct the coefficients from initial and final conditions. The behavior of the algorithm has been studied for both exact and perturbated measurements. First results are presently obtained during discharges of the European tokamak TORE SUPRA seated in Cadarache (France).

5. CONCLUSION

Those preliminary results show that the method allows to identify transport coefficients in a way consistent with full 2D equilibrium and experimental data. Great improvements in understanding transport phenomena are made possible and it remains to rely those coefficients to a global physical theory.

Acknowledgement : the authors are grateful to J. Le Foll who is at the origin of this work for many helpful discussions about this problem.

REFERENCES

[1] J. BLUM : Numerical Simulation and Optimal Control in Plasma Physics, Wiley/Gauthiers-Villars, 1989.

[2] J. BLUM et al. . Problems and methods of self-consistent reconstruction of tokamak equilibrium profiles from magnetic and polarimetric measurements, Nuclear Fusion 30, 8, 1990.

[3] J. BLUM, Y. STEPHAN : Identification of the plasma current density in a tokamak, 5th IFAC Symposium, Perpignan 1989.

[4] J. L. LIONS : Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles, Dunod 1968.

STABILITY PROPERTIES AND ASYMPTOTIC STATES OF A 2D TEARING-UNSTABLE PLASMA

B.SARAMITO and E.K.MASCHKE

Université Clermont-Ferrand II
Département de Mathématiques
F-63177 Aubière, Cédex (France)

Assoc. EURATOM-CEA FUSION
D.R.F.C., C.E.N. Cadarache
F-13108 St-Paul-lez-Durance, Cédex

Abstract - We investigate numerically the nonlinearly saturated single-helicity tearing instability of a visco-resistive current-carrying plane plasma slab using a 2D spectral code. The ratio of viscosity to resistivity is fixed ($\nu/\eta = 0.2$). The equilibrium state depends on the x-coordinate only, perturbations are supposed to be periodic in the y-direction (period L) and independent of the z-direction. The Lundquist number S is chosen as bifurcation parameter, and we investigate solutions with different fixed values of the period L by varying S.

Choosing a sufficiently low value of L ($L = L_0$), the first branch of the set of solutions bifurcating from the given static equilibrium is numerically found to be stable up to high values of the bifurcation parameter ($S = 10^6$). Passing to a new value $L = 2L_0$, that same branch presents a symmetry breaking. For a period $L = 4L_0$, that branch becomes unstable but a lower branch is found stable. Moreover, depending on the choice of the initial conditions, the evolution code may yield spatially very complicated transient states.

I. THE 2D TEARING INSTABILITY

We briefly recall here the physical and mathematical model [1],[2]. Let $\Omega =]-1/2, +1/2[\times]0, L[$ be an open set in $\mathbb{R}^2$. The unknowns ψ, ϕ, ω obey the following equations:

$$\frac{\partial\psi}{\partial t} - \Delta\psi + S\left(V.\nabla\psi + V_x \frac{d\psi_{eq}}{dx}\right) = 0$$

$$\Delta\phi = \omega$$

$$\frac{\partial\omega}{\partial t} - \frac{\nu}{\eta}\Delta\omega + S\left(\frac{\partial(\phi,\omega)}{\partial(y,x)} - \frac{\partial(\psi,\chi)}{\partial(y,x)} + \frac{d\psi_{eq}}{dx}\frac{\partial\chi}{\partial y} - \frac{d^3\psi_{eq}}{dx^3}\frac{\partial\psi}{\partial y}\right) = 0$$

where $\chi = \Delta\psi$, and ψ_{eq} is a given static equilibrium magnetic flux such that

$$\psi_{eq}(x) = -(1/\alpha)\log\{ch(\alpha x)\}.$$

ν and η are the (constant) viscosity and resistivity, respectively ($\nu/\eta = 0.2$), and S is the Lundquist number (bifurcation parameter).

The velocity and the magnetic field are related to ϕ and ψ as follows:

$V = \nabla\phi \times ez$, $B = \nabla\psi \times ez$.

We choose periodic boundary conditions in the y direction (sometimes imposing an additional symmetry) and take

$\phi = \Delta\phi = \psi = 0$ at $x = \pm 1$.

For the numerical calculations, χ is also an unknown of the problem, with $\chi = 0$ at $x = \pm 1$.

As already proved [2], we can mathematically justify the existence of bifurcation, making use of compact operators.

The equilibrium magnetic field corresponding to the flux function ψ_{eq}, given above, is parallel to the y axis and changes sign at $x = 0$. When the instability sets in, a magnetic island appears (see the representations of curves $\psi(x,y)$ = constant in the figure).

For the problem of linear stability of the equilibrium we look for solutions expanded in Fourier series.

$\exp(\omega t + imky)$ with $m \in \mathbb{N}$ (1)

$k = 2\pi/L$ (or π/L if we impose symmetries in the y-direction).

A pseudo-spectral code is used to solve our equations, with Fourier-Galerkin decomposition in y, and Chebyshev-tau approximation in x, together with implicit or semi-implicit discretization schemes in time. We also impose symmetries in the two directions x and y.

II. BRANCHES OF STATIONARY SOLUTIONS

II. 1. Structure of the stationary solutions

We consider three different values of the parameter k, namely : $k = 2.5$, $k = 1.25$, $k = 0.625$, corresponding respectively to lengths L, 2L, 4L in the y-direction.

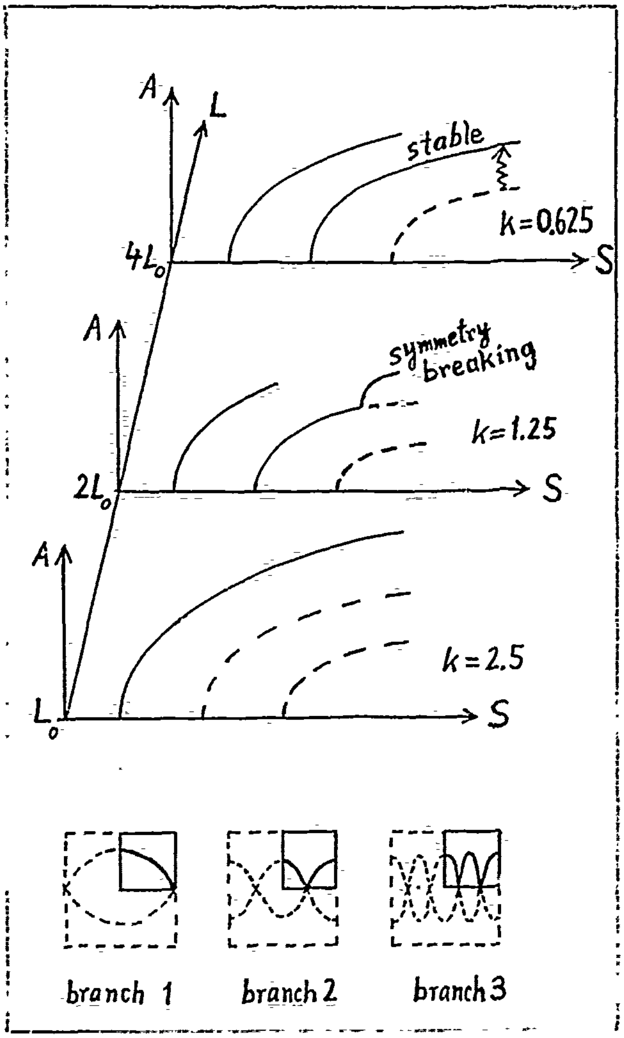
To each number m of formula (1) we associate a value S(m) of the parameter S, which corresponds to a bifurcation point where a branch of nonlinear stationary solutions (labelled as "branch m") bifurcates from the equilibrium.

Our main results are represented in the figure and may be summarized as follows:

For the length Lo (k = 2.5), the branch 1 has been found to be numerically stable. Taking into account the imposed symmetries, our computation yields only a part of the magnetic island as shown schematically at the bottom of the figure (the lines inside each box represent the separatrix of the magnetic field configuration).

The branch 2 for length 2Lo (obtained by symmetry in the y-direction from the branch 1 corresponding to length Lo) is again numerically stable, but it exhibits a symmetry breaking.

For length 4Lo, the branch 3 (obtained by symmetry from the branch 2 corresponding to length 2Lo) is now found to be numerically unstable, with solutions converging in time towards the (stable) solution of the branch 2 of length 4Lo.



II. 2. Sensitivity to initial conditions

Choosing some initial conditions in the vicinity of a stationary stable solution, we have sometimes observed periodic oscillations in time, with a very slowly decreasing amplitude. Moreover, if we start away from a given stationary stable solution, the evolution of the system presents numerically certain spatially very complicated transient states. These phenomena will be investigated in later work.

References

[1] E.K.Maschke and B.Saramito, Turbulence and transport associated with saturated tearing modes. In: "Turbulence and Anomalous Transport in Magnetized Plasmas", D.Gresillon and M.Dubois Eds., Editions de Physique, Orsay (France), 1987.

[2] B.Saramito, Thèse d'Etat, Université Paris VI, 1987.

HIERARCHIC FINITE ELEMENTS FOR MINDLIN PLATES

LUCIA DELLA CROCE
Dipartimento di Matematica
Università di Pavia
I-27100 Pavia, Italia

and

TERENZIO SCAPOLLA
Dipartimento di Matematica
Università di Pavia
I-27100 Pavia, Italia

Abstract

The Reissner-Mindlin model describes the deformation of a plate subiect to a transverse loading in terms of the normal displacement of the midplane and the rotation of the fibers normal to the midplane. T e model is widely used for plates of small to moderate thickness.

It is well known that the numerical approximation of the Reissner-Mindlin plate with standard low degree finite elements leads to solutions that are very sensitive to the plate thickness. For small thickness the numerical solution is very far from the exact one. The phenomenon is referred to as *locking.* Several non-standard finite element spaces and techniques have been devised to overcome such a difficulty. We recall, among others, reduced and selective integration, interpolation and projection techniques combined with non-classical formulations like mixed or hybrid approaches. As far as we know only low order element, generally with degree 1 or 2, have ben tested for Reissner-Mindlin plates, both for standard and non-standard formulations.

In recent years high order finite elements, known as *p* finite elements, have been introduced and succesfully applied in several fields, e.g., for elasticity and Kirchhoff plate problems. High order elements have shown to be robust and able to absorb locking phenomena in the case of nearly incompressible materials. In this note we present some numerical results obtained with a family of *hierarchic* finite elements with degree from 1 to 4 for the solution of Reissner-Mindlin plates in the standard formulation. We show that the locking of the solution does not appear when high order elements are used.

1. Derivation of the Reissner-Mindlin model

Let us consider a three-dimensional body occupying a region V. We assume that the body under consideration is isotropic. The potential (or strain) energy per unit volume is given by

$$W = \frac{1}{2}\lambda\left(e_{xx} + e_{yy} + e_{zz}\right)^2$$

$$+\mu\left(e_{xx}^2 + e_{yy}^2 + e_{zz}^2 + 2e_{yz}^2 + 2e_{zx}^2 + 2e_{xy}^2\right)$$

where $e_{xx}, e_{yy}, e_{zz}, e_{yz}, e_{zx}, e_{xy}$ are the components of strain and λ and μ are the Lamé coefficients, constant characterizing the elastic behaviour of the body. The coefficients can be expressed in terms of the modulus of elasticity E and the Poisson's ratio ν:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

The components of stress at a given point are linear and homogeneous functions of the components of strain at the same point and viceversa (see, e.g., [5]). Denoting by u, v, w the components of displacement we have the relations between strain and displacement:

$$e_{xx} = \frac{\partial u}{\partial x}$$

$$e_{yy} = \frac{\partial v}{\partial y}$$

$$e_{zz} = \frac{\partial w}{\partial z}$$

$$e_{yz} = \frac{1}{2}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)$$

$$e_{zx} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)$$

$$e_{xy} = \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)$$

In the Reissner-Mindlin theory two new fields ϕ_1 and ϕ_2 are introduced. These fields represent the rotation of the cross-sectional planes to which, respectively, the x-axis and y-axis is normal:

$$u = -z\phi_1(x,y)$$

$$v = -z\phi_2(x,y)$$

$$w = w(x,y)$$

Through the Reissner-Mindlin hypotheses the total strain energy becomes:

$$U(w,\phi_1,\phi_2)$$

$$= \tfrac{1}{2}\frac{Et^3}{12(1-\nu^2)}\int_\Omega \Big(\phi_{1/x}^2 + \phi_{2/y}^2 + 2\nu\phi_{1/x}\phi_{2/y}$$

$$+ \frac{1-\nu}{2}(\phi_{1/y} + \phi_{2/x})^2\Big)\,dxdy$$

$$+ \tfrac{1}{2}\frac{Etk}{2(1+\nu)}\int_\Omega \left((w_{/x} - \phi_1)^2 + (w_{/y} - \phi_2)^2\right)dxdy$$

where k is the shear correction factor.

Taking into account the external load p we get the following variational formulation for the Reissner-Mindlin model for plates:

$$
\begin{aligned}
&U(w;\phi_1,\phi_2;v,\psi_1,\psi_2)\\
&=\frac{Et^3}{12(1-\nu^2)}\int_\Omega\Big(\phi_{1/x}\psi_{1/x}+\phi_{2/y}\psi_{2/y}\\
&+\nu\left(\phi_{1/x}\psi_{2/y}+\phi_{2/y}\psi_{1/x}\right)\\
&+\frac{1-\nu}{2}(\phi_{1/y}+\phi_{2/x})(\psi_{1/y}+\psi_{2/x})\Big)\,dxdy\\
&+\frac{Etk}{2(1+\nu)}\int_\Omega\Big((\phi_1-w_{/x})(\psi_1-v_{/x})\\
&+(\phi_2-w_{/y})(\phi_2-vy)\Big)\,dxdy-\int_\Omega pv\,dxdy
\end{aligned}
$$

Setting $\bar\phi=(\phi_1,\phi_2)$ the previous problem can be stated in the following way:

$$
\begin{cases}
\text{Find } (\bar\phi,w)\in\left(H_0^1\right)^2\times H_0^1 \text{ such that}\\
\dfrac{Et^3}{12(1-\nu^2)}\mathrm{A}(\bar\phi,\bar\psi)+\dfrac{Etk}{2(1+\nu)}\|\nabla w-\bar\phi\|_0^2=(p,v)\\
\forall(\bar\psi,v)\in\left(H_0^1\right)^2\times H_0^1
\end{cases}
$$

where

$$
\begin{aligned}
\mathrm{A}(\bar\phi,\bar\psi)=\int_\Omega\Big(&\phi_{1/x}\psi_{1/x}+\phi_{2/y}\psi_{2/y}\\
&+\nu\left(\phi_{1/x}\psi_{2/y}+\phi_{2/y}\psi_{1/x}\right)\\
&+\frac{1-\nu}{2}(\phi_{1/y}+\phi_{2/x})(\psi_{1/y}+\psi_{2/x})\Big)\,dxdy.
\end{aligned}
$$

2. Numerical approximation

One of the advantages of the Reissner-Mindlin approach is the fact that the variational formulation allows the use of continuos (C^0) finite elements, since only first derivatives appear. We recall that in the Kirchhoff model conforming approximations require C^1 finite elements. It is well known that the numerical approximation of the Reissner-Mindlin plate with standard low degree finite elements leads to solutions that are very sensitive to the plate thickness. For small thickness the numerical solution is very far from the exact one. The phenomenon is referred to as *locking*. Several non-standard finite element spaces and techniques have been devised to overcome such a difficulty. We recall, among others, reduced and selective integration, interpolation and projection techniques combined with non-classical formulations like mixed or hybrid approaches. As far as we know only low order element, generally with degree 1 or 2, have ben tested for Reissner-Mindlin plates, both for standard and non-standard formulations.

In recent years high order finite elements, known as p finite elements, have been introduced and succesfully applied in several fields, e.g., for elasticity and Kirchhoff plate problems (see [1,2,3]). High order elements have shown to be robust and able to absorb locking phenomena in the case of nearly incompressible materials (see [4]). In this whay we present some numerical results obtained with a family of *hierarchic* finite elements for the Reissner-Mindlin plate in the standard formulation. Hierarchy of finite elements means that shape functions are added to increase the degree of approximation, leaving unchanged the previous functions. The functions used are based on the family of Legendre polynomials. This class presents good properties from the point of view of roundoff error accumulation whith respect to the increase of the polynomial degree. The family consists of finite elements of degree from 1 to 4.

Referring to the classification of the shape functions suggested by Babuška [1] as *nodal*, *side* and *internal* functions, we give in Table 1 the number of sha, e functions of each type for general value of the degree p.

d.	*nodal* f.	*side* f.	*internal* f.	total # f.
1	4	–	–	4
2	4	4	–	8
3	4	8	–	12
4	4	12	1	17
5	4	16	3	23
6	4	20	6	30
⋮	⋮	⋮	⋮	⋮
p	4	$4(p-1)$	$\frac{1}{2}(p-2)\times$ $(p-3)$	$4p$ $+\frac{1}{2}(p-2)(p-3)$

Table 1: Number of shape functions for different degrees

We have written a code where shape functions with degree from 1 to 4 are used. The hierarchic structure of the functions allows easy extensions to higher degrees.

We introduce two finite dimensional spaces Φ and V, respectively for the discrete rotations and displacement. The approximate problem can be stated in the following way:

$$
\begin{cases}
\text{Find } (\bar\phi^h,w^h)\in\Phi\times V \text{ such that}\\
\dfrac{Et^3}{12(1-\nu^2)}\mathrm{A}(\bar\phi^h,\bar\psi^h)+\dfrac{Etk}{2(1+\nu)}\|\nabla w^h-\bar\phi^h\|_0^2=(p,v^h)\\
\forall(\bar\psi^h,v^h)\in\Phi\times V
\end{cases}
$$

where

$$
\begin{aligned}
\mathrm{A}(\bar\phi^h,\bar\psi^h)=\int_\Omega\Big(&\phi^h_{1/x}\psi^h_{1/x}+\phi^h_{2/y}\psi^h_{2/y}\\
&+\nu\left(\phi^h_{1/x}\psi^h_{2/y}+\phi^h_{2/y}\psi^h_{1/x}\right)\\
&+\frac{1-\nu}{2}(\phi^h_{1/y}+\phi^h_{2/x})(\psi^h_{1/y}+\psi^h_{2/x})\Big)\,dxdy.
\end{aligned}
$$

3. Numerical results

A series of numerical experiments on several plates with different physical data has been performed. Due to the lack of space only a few of them are presented. Comparison are made with the theoretical results computed by Timoshenko [6].

For each test, among others, displacement at the center C of the plate and the strain energy have been computed. Let $w(C)$ denote the exect displacement at the center of the plate and $w_h(C)$ the finite element solution. The relative displacement error is defined as

$$d = \frac{w(C) - w_h(C)}{w(C)} \times 100.$$

The exact strain energy was not available. Out of the discrete strain energy an extrapolation has been made in order to get an accurate value of the energy. Let E denote such a energy, let E_h be the discrete energy. The relative energy norm $\|e\|$ of the error $e = w - w_h$ can be expressed in the following way:

$$\|e\| = \left(\frac{E - E_h}{E}\right)^{1/2} \times 100.$$

We consider a square plate with uniform decomposition. On the boundary the plate is clamped. Due to the simmetry we have solved the problem on a quarter of plate. We have considered a unit plate with thickness 0.1 (thick plate), 0.01, 0.001, 0.0001. The last value is related to an ultrathin plate. We have considered this value in order to assess the stabilit that the locking is larger when small thicknesses are taken into account.

In Fig.1a–d we consider a plate with thickness $t = 0.01$. The relative error for the displacement *vs.* number of degrees of freedom is shown. The results are given for values of the degree $p = 1, 2, 3, 4$.

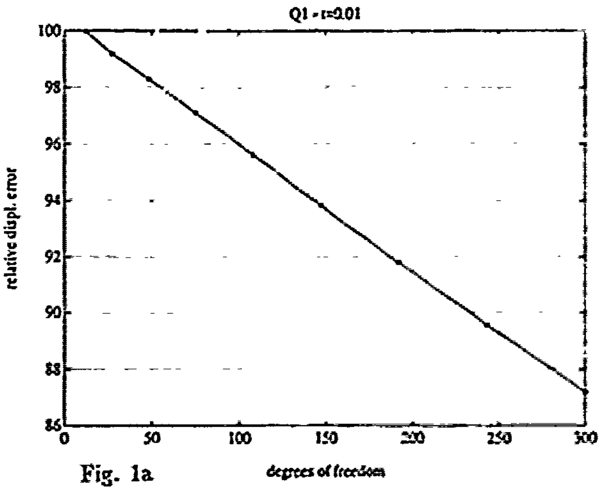


Fig. 1a

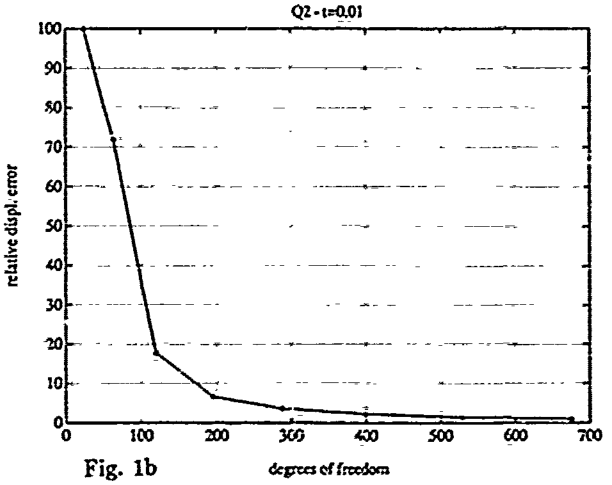


Fig. 1b

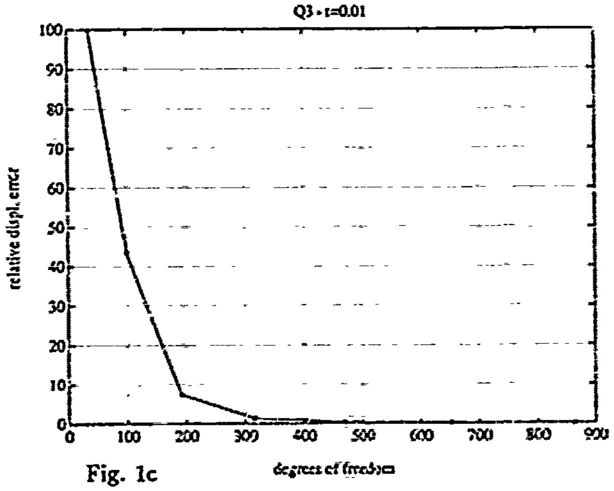


Fig. 1c

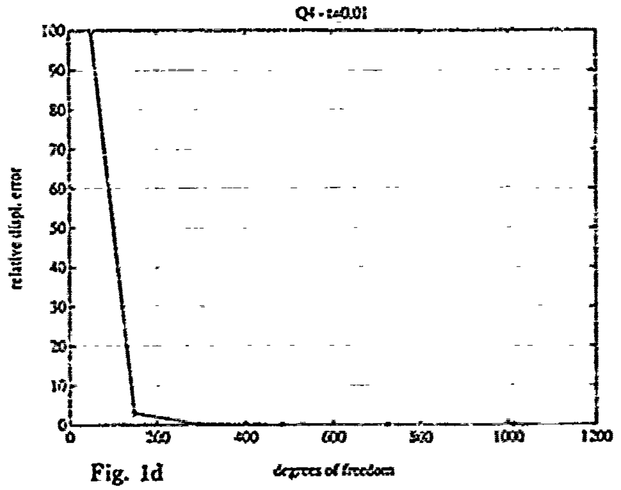


Fig. 1d

The previous results are related to a moderately thick plate. The pictures show that for $p = 1$ the rate of convergence is very slow and the solution is very inaccurate. The numerical solution exhibits a strong locking. For $p = 2,3,4$ we note a good behaviour. The respective displacement solutions approach the exact solution with a relatively small number of degrees of freedom. As expected, the better convergence is achieved for increasing values of p.

In Fig.2a–d we consider a plate with thickness 0.001. We show, in log–log scale, the relative energy norm error *vs.* number of degrees of freedom. The results are given for values of the degree $p = 1, 2, 3, 4$.

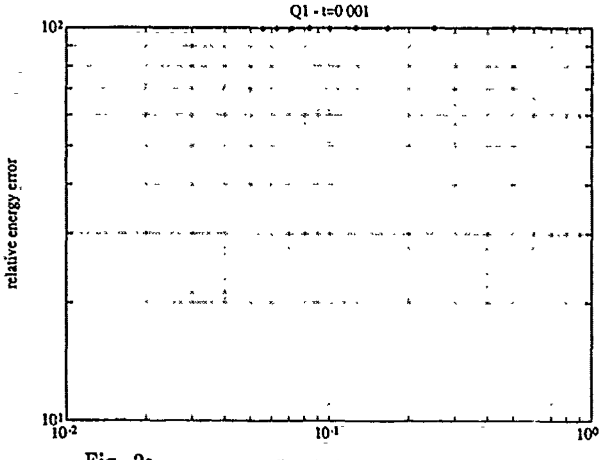


Fig. 2a

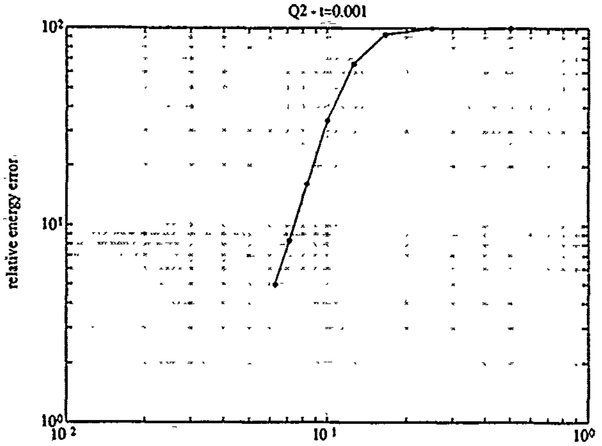


Fig. 2b

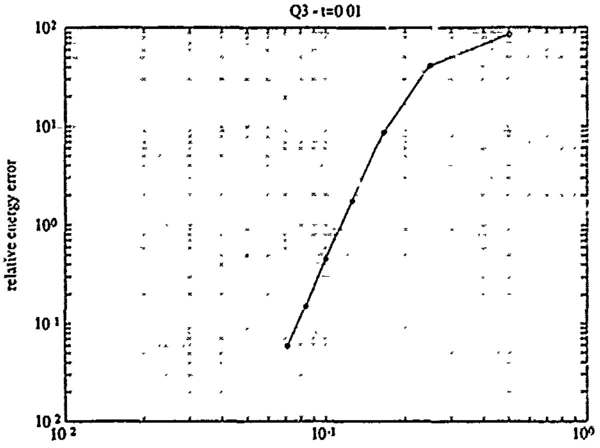


Fig. 2c

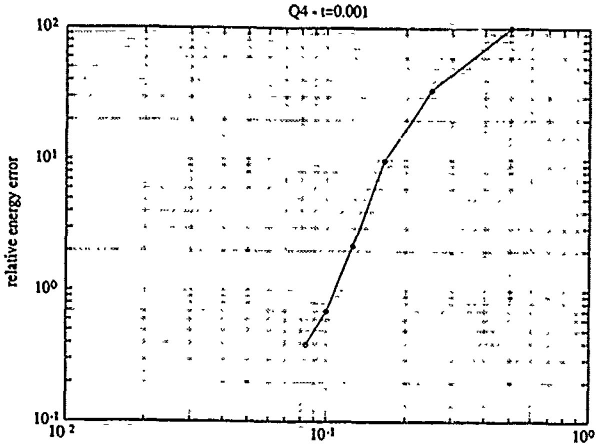


Fig. 2d

In the previous pictures a thin plate is considered. The discrete strain energy is a good global indicator since it takes into account a distributed solution. For $p = 1$ we observe no convergence. The behaviour is far worst than the case with thickness $t = 0.01$.

In Fig.3a–d we consider, for a fixed degree of approximation $p = 2$, different values of the thickness. We show, in log–log scale, the relative error for the displacement *vs.* number of degrees of freedom. freedom.

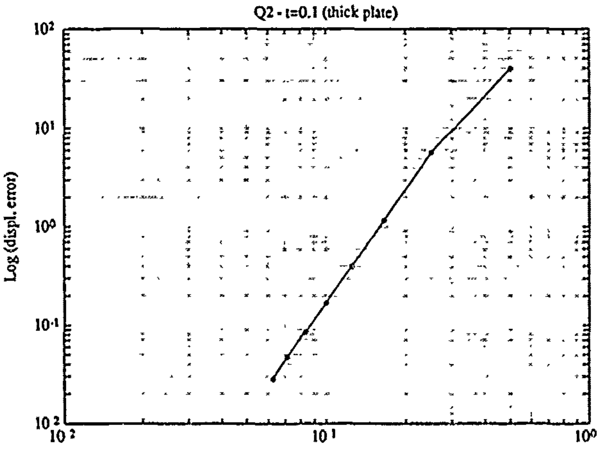


Fig. 3a

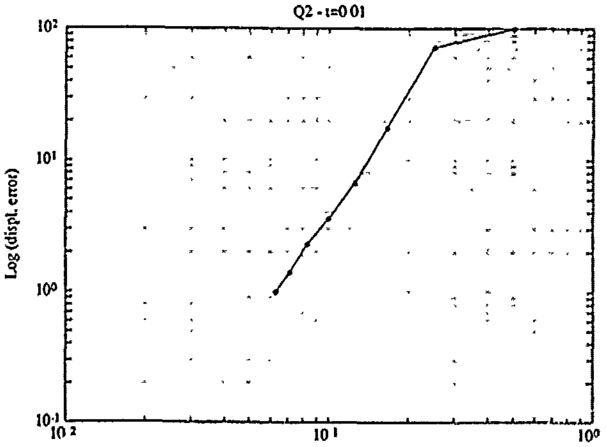


Fig. 3b

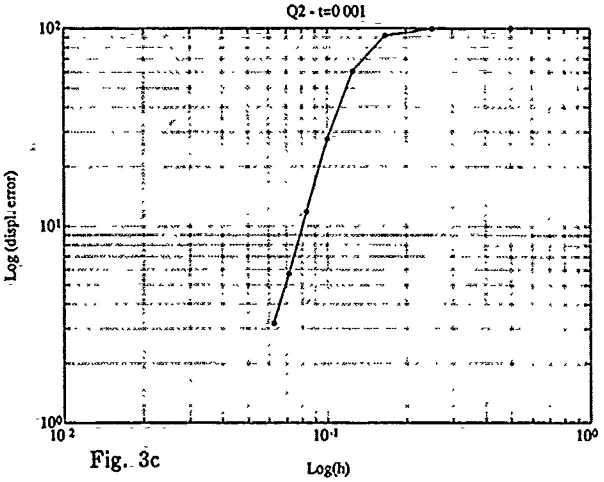


Fig. 3c

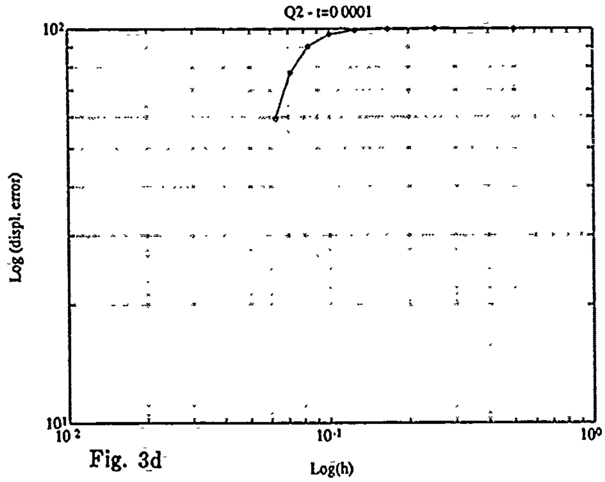


Fig. 3d

In the previous pictures we consider the numerical approximation with fixed degree $p = 2$ and we change the value of thickness. The sequence of figures well shows that the convergence slows down noticeably when the thickness is reduced. The log-log scales allow to evaluate the different rates of convergence (note the different scaleson the y-axis). In particular, for the smaller thickness, the locking of the numerical solution is well exhibited.

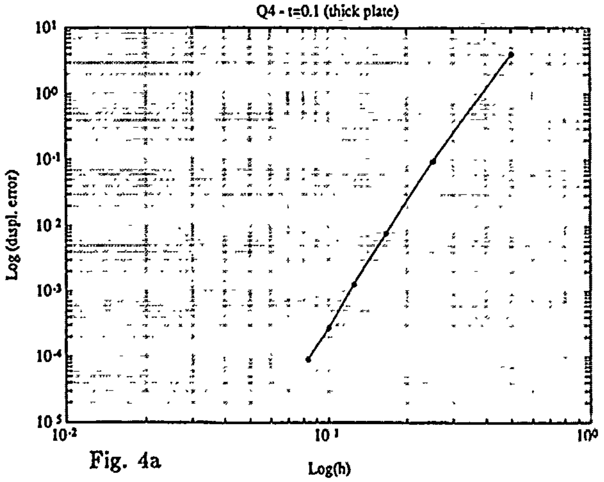


Fig. 4a

In Fig.4a–d we consider, for a fixed degree of approximation $p = 4$, different values of the thickness. We show, in log-log scale, the relative error for the displacement *vs.* number of degrees of freedom. freedom.

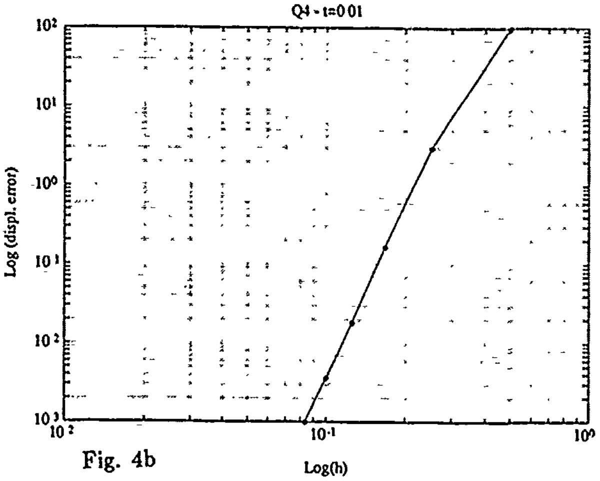


Fig. 4b

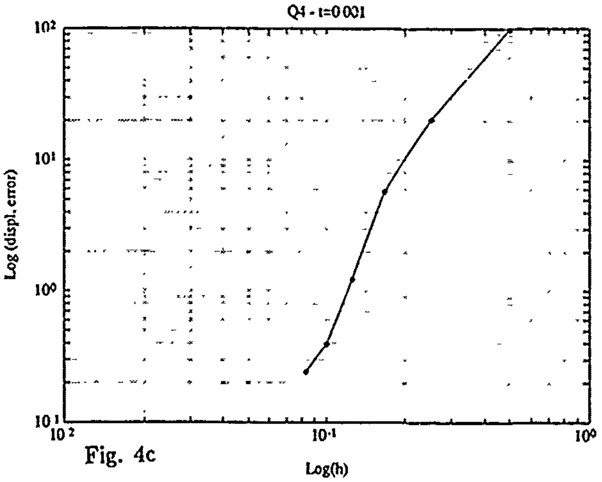


Fig. 4c

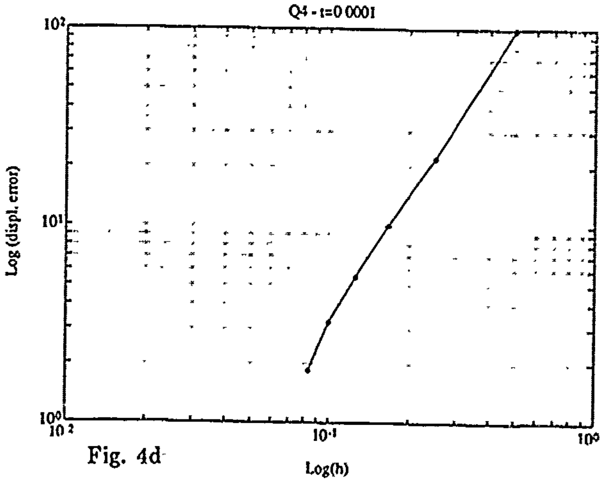


Fig. 4d

We now consider a fixed degree $p = 4$ and we changethe thickness. We observe that, with such an high order element, convergence is achieved even for the smaller value of thickness $t = 0.0001$ This shows that the high order element is able to to absorb the phenomenon of locking found for lower degrees (see Fig.3a–d).

[1] Babuška I.: The *p* and *h-p* versions of the finite element method. The state of the art, in *Finite Elements – Theory and Application*, D.L. Dwoyer, M.Y. Hussaini, R.G. Voigt (eds), Springer-Verlag, New York, 199–239 (1988)

[2] Babuška I., Scapolla T.: Benchmark computation and finite element performance for a rhombic plate bending problem, *International J. for Numerical Methods in Engineering*, 28, 155–179 (1989)

[3] Chinosi C., Sacchi G., Scapolla T.: Hierarchic conforming finite elements for plate bending problems, to appear on *Computational Mechanics*

[4] Szabo B.A., Babuška I., Chayapathy K.: Stress Computations for Nearly Incompressible Materials by the *p*-Version of the Finite Element Method, *International J. for Numerical Methods in Engineering*, 28, 2175–2190 (1989)

[5] Muskhelishvili N.I.: *Some Basic Problems of the Mathematical Theory of Elasticity*, Noordhoff, Groningen (1963)

[6] Timoshenko S., Woinowski-Krieger S.: *Theory of Plates and Shells*, McGraw Hill, Singapore (1970)

VISUALIZATION OF ELASTIC WAVE

YAN ZHAO
Dept. of computer science
Arizona State University
Tempe, AZ 85287

and

ROSEMARY RENAUT
Dept. of Mathematics
Arizona State University
Tempe, AZ 85287

Abstract. A finite difference method is used to solve the system of 2-D elastic wave equations. The solution is expressed as the displacement of a particle from its original position. By means of computer graphics we describe the solution as a 3-D wave surface, and describe its propogation by real time animation. In this way, we can observe the phenomena described by the system of equations more easily and in a much more obvious fashion than with conventional methods. Here, we use an explicit method to solve the equations in time, this makes real time calculation, modeling and rendering possible. The algorithm has been implemented on an Iris-4D personal graphics workstation.

1. Introduction

Computer graphics provides an excellent means for exploring the elastic waves visually. Here we shall give an experimental procedure for modeling and rendering the elastic wave and its propogation in real time. To make things easier for demonstration we choose some typical parameters for the wave equation. That will not have any influence on the method to express the solutions.

Mainly, we discuss solutions obtained from second order finite differences. We suppose that at begining the wave surface looks like a plane, nothing happens until an external body force is applied. The external body force car act at any time, last any period and be at any place. The source function can be given in any form you want. We suppose two kinds of boundary conditions . 1> a free boundary condition, 2> a fixed boundary condition.

2. 2-D elastic wave equations

The equation is written as a system of second order wave equations for the displacement vector U(x,t) = [U(x,t), V(x,t)], see [1]. Here U(x,t) describes the horizontal displacement , V(x,t) the vertical displacement, and x=(x, y) within the domain of interest.

$$pU_{tt} = [c(U_x + V_y) + 2wU_x]_x + [w(U_y + V_x)]_y + pf1$$
$$pV_{tt} = [w(U_y + V_x)]_x + [c(U_x + V_y) + 2wV_y]_y + pf2 \qquad (1)$$

The Lame parameters c=c(x) and w=w(x) as well as the density p=p(x) can vary in space. The external body forces are denoted by the source functions f1(x,t) and f2(x,t). If we substitute

$$f = c(U_x+V_y) + 2wU_x$$
$$g = w(U_y+V_x) \qquad (2)$$
$$h = c(U_x+V_y) + 2wV_y ,$$

the equations can be written as

$$pu_t = f_x + g_y + pf1$$
$$pv_t = g_x + h_y + pf2$$
$$f_t = (c+2w)u_x + cv_y \qquad (3)$$
$$gt = w(v_x+u_y)$$
$$ht = cu_x + (c+2w)u_y .$$

In these equations, (u,v) = (Ut,Vt) represents the velocity of the material particles.

3. Finite-difference method

Here we use the 2nd order approximation for the first derivatives.

$$u'(x_0) = [\ u(x_0+h) - u(x_0-h)\ ] / 2h \qquad (4)$$

Let

$$u^n_{lm} = u\ (l\,\Delta x,\ m\,\Delta y,\ n\,\Delta t),\ \ \Delta x = \Delta y = h,\ \ \Delta t = k,\ \ r = k/h$$

$$0 < l < L,\ \ 0 < m < M,\ \ ' > 0 .$$

Then (3) can be expressed as an explicit system

$$u^{n+1}_{lm} = u^{n-1}_{lm} + (r/p)[f^n_{l+1\ m} - f^n_{l-1\ m} + g^n_{l\ m+1} - g^n_{l\ m-1}] + 2k\,f_1(nk)$$
$$v^{n+1}_{lm} = v^{n-1}_{lm} + (r/p)[g^n_{l+1\ m} - g^n_{l-1\ m} + h^n_{l\ m+1} - h^n_{l\ m-1}] + 2k\,f_2(nk)$$
$$f^{n+1}_{lm} = f^{n-1}_{lm} + r\,(e+2w)(u^n_{l+1\ m} - u^n_{l-1\ m}) + r\,e\,(v^n_{l\ m+1} - v^n_{l\ m-1})$$
$$g^{n+1}_{lm} = g^{n-1}_{lm} + w\,r\,[v^n_{l+1\ m} - v^n_{l-1\ m} + u^n_{l\ m+1} - u^n_{l\ m-1}]$$
$$h^{n+1}_{lm} = h^{n-1}_{lm} + e\,r\,(u^n_{l+1\ m} - u^n_{l-1\ m}) + (e+2w)(v^n_{l\ m+1} - v^n_{l\ m-1}) \qquad (5)$$

Using the Trapezoidal rule, we get the displacement U and V as

$$U^{n+1}_{lm} = U^n_{lm} + 0.5k\,(u^{n+1}_{lm} + u^n_{lm})$$
$$V^{n+1}_{lm} = V^n_{lm} + 0.5k\,(v^{n+1}_{lm} + v^n_{lm}) \qquad (6)$$

Here we should observe that the solution points of all the functions U, V, u, v, f, g, h are uncoupled in space domain, but using this method they can be divided into two coupled sets of points. This means that the solution for each function has been separated into two independent groups along each axis, the group of even points and the group of odd points. The value at one even point only affects other even points. The value at one odd point only affects other odd points For the propagation of the wave, this phenomenon means that if an external body force acts at an even point it will only propagate among the even points. Equation (5) and Diagram 1 can explain why this happens to u and v. The equations for u and v given by (5) in the x direction are considered here. We suppose that the u wave begins at u_{lm} and the v wave begins at v_{lm} , then they affect $f_{l+1\ m}$, $f_{l-1\ m}$, $g_{l+1\ m}$, $g_{l-1\ m}$, $h_{l+1\ m}$, $h_{l-1\ m}$ at the next time step After two time steps the waves only propagate to u_{lm} , $u_{l+2\ m}$, $u_{l-2\ m}$ and v_{lm} , $v_{l+2\ m}$, $v_{l-2\ m}$, nothing will be affected at $u_{l+1\ m}$, $u_{l-1\ m}$ and $v_{l+1\ m}$, $v_{l-1\ m}$. From equation (6), we see that functions U and V are the ODE solutions of the function u and v in time, and thus this uncoupled property is also observed in U and V. However, we can still use these kind of solutions to demonstrate our visualization technique by taking only the even points or only the odd points, depending on where the external body force acts. But observe thet because of this uncoupling, the Courant Fredrichs Lewy condition is not satisfied and thus these solution are not truely physical [3]. We will discuss this in more detail in a later paper.

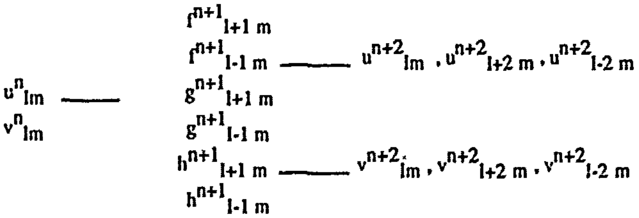


Diagram 1

If a higher accuracy solution is desired, an approximation for the first derivative of order 4 can be used, see [2]. The formula is given by

$$u'(x_0)=[-(1/6)u(x_0+2h)+(4/3)u(x_0+h)-(4/3)u(x_0-h)+(1/6)u(x_0-2h)]/2h \qquad (7)$$

This will exhibit almost the same uncoupled properties as formula (5).

Initial values and boundary conditions can be chosen as desired. Here, initial values are chosen as zero. Two kinds of boundary conditions are chosen here. One is the free boundary condition, or zero Neumann boundary condition. The other the fixed boundary condition, the usual zero Dirichlet boundary condition.

Initial values:

$$\text{At } t=0 \text{ and } t=k: \quad U=V=u=v=f=g=h=0. \qquad (8)$$

Boundary conditions:

A. Free boundary:

$$U^n_{l\,M}=U^n_{l\,M-1}, \quad U^n_{l\,0}=U^n_{l\,1}, \quad V^n_{l\,M}=V^n_{l\,M-1}, \quad V^n_{l\,0}=V^n_{l\,1},$$
$$U^n_{L\,m}=U^n_{L-1\,m}, \quad U^n_{0\,m}=U^n_{1\,m}. \quad V^n_{L\,m}=V^n_{L-1\,m}, \quad V^n_{0\,m}=V^n_{1\,m}.$$

$$u^n_{l\,M}=u^n_{l\,M-1}, \quad u^n_{l\,0}=u^n_{l\,1}, \quad u^n_{L\,m}=u^n_{L-1\,m}, \quad u^n_{0\,m}=u^n_{1\,m};$$
$$v^n_{l\,M}=v^n_{l\,M-1}, \quad v^n_{l\,0}=v^n_{l\,1}, \quad v^n_{L\,m}=v^n_{L-1\,m}, \quad v^n_{0\,m}=v^n_{1\,m};$$
$$f^n_{l\,M}=f^n_{l\,M-1}, \quad f^n_{l\,0}=f^n_{l\,1}, \quad f^n_{L\,m}=f^n_{L-1\,m}, \quad f^n_{0\,m}=f^n_{1\,m};$$
$$g^n_{l\,M}=g^n_{l\,M-1}, \quad g^n_{l\,0}=g^n_{l\,1}, \quad g^n_{L\,m}=g^n_{L-1\,m}, \quad g^n_{0\,m}=g^n_{1\,m};$$
$$h^n_{l\,M}=h^n_{l\,M-1}, \quad h^n_{l\,0}=h^n_{l\,1}, \quad h^n_{L\,m}=h^n_{L-1\,m}, \quad h^n_{0\,m}=h^n_{1\,m}. \qquad (9)$$

B. Fixed boundary:

$$U^n_{l\,0}=U^n_{l\,M}=0, \quad V^n_{l\,0}=V^n_{l\,M}=0,$$
$$U^n_{0\,m}=U^n_{L\,m}=0. \quad V^n_{0\,m}=V^n_{L\,m}=0.$$
$$u^n_{l\,0}=u^n_{l\,M}=0, \quad u^n_{0\,m}=u^n_{L\,m}=0;$$
$$v^n_{l\,0}=v^n_{l\,M}=0, \quad v^n_{0\,m}=v^n_{L\,m}=0;$$
$$f^n_{l\,0}=f^n_{l\,M}=0, \quad f^n_{0\,m}=f^n_{L\,m}=0;$$
$$g^n_{l\,0}=g^n_{l\,M}=0, \quad g^n_{0\,m}=g^n_{L\,m}=0;$$
$$h^n_{l\,0}=h^n_{l\,M}=0, \quad h^n_{0\,m}=h^n_{L\,m}=0. \qquad (10)$$

4. Parameter selection

Actually, we can select any parameter, source function and area of interest that we wish to observe. As some simple examples, the parameters are selected as p(x,y) = e(x,y) = w(x,y) = 1.The external body force is selected as

1>

$$f1 = 0,$$
$$f2 = \delta(x-x_0)\,\delta(y-y_0)\,S_1(t). \qquad (11)$$

2>

$$f1 = \delta(x-x_0)\,\delta(y-y_0)\,S_1(t),$$
$$f2 = \delta(x-x_0)\,\delta(y-y_0)\,S_2(t). \qquad (12)$$

Here

$$S_1(t)=\begin{cases}A\sin(4\pi t) & 0<t-t_s<t_s\\ 0 & t_s<t\end{cases}, \qquad S_2(t)=\begin{cases}A\cos(4\pi t) & 0<t-t_s<t_s\\ 0 & t_s<t\end{cases}$$

5. Rendering

From the solution of the above difference equation system, we can get the discretized points of the wave surfaces described by U(lh,mh,nk) and V(lh,mh,nk). Also, we can get the solution of the surfaces described by the speed functions u(lh,mh,nk) and v(lh,mh,nk). As the solutions are uncoupled, along each axis, we take only the even points or only the odd points to form each solution mesh, depending on whether the external body forces act at an even point or an odd point.

For surface rendering, we should first get the normal for each vertex of the solution mesh. This can be done by taking the average of the normals on the four neighbouring triangular patches, as the normal for each neighbouring triangular patch can be easily obtained. This is shown in figure 1.

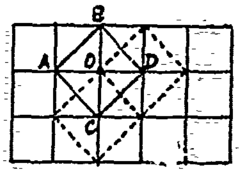


Fig. 1

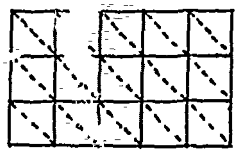


Fig. 2

The triangulated surface is then obtained by joining one pair of diagonal points for each square of the solution mesh. This is shown in figure 2. Phong's lighting model is used for getting color at each vertex of the solution mesh and Gouraud shading is used for each triangular patch rendering on the surface. The double buffer is used for rendering, for a better results in continuous cha:

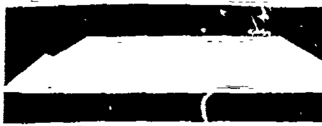
Because we use an explicit method for the time variable t, the time spent for dealing with the equations allows for real time animation.

6. Example

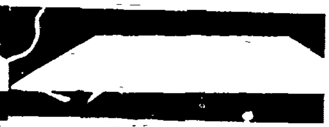
The area of interest chosen here is $0 < x < 1, 0 < y < 1$, so that Mh = Lh = 1, and t>0. Chose M = L = 20, h = 0.05, k = 0.008, $x_0 = y_0 = 0.1$, $t_s = 0.2$. Fixed boundary condition (10) and external body force function (11) are used. Pictures 1 - 8 show the vertical displacement function V(x, t) at the times for n = 16, 48, 64, 80, 112, 160, 192, 256. Here we see that the wave begins from the lower left corner and propagates towards the upper right corner.

7. Conclusion

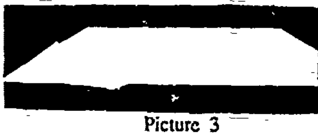
We have shown a way to visualize elastic wave propogation with real time animation by means of computer graphics. This improves interpretation of the numerical results. We can use this method to show how the different media or different external forces affect on the elastic waves and how elastic waves propogate, as well as how different parameters, boundary conditions or initial conditions affect the solutions.



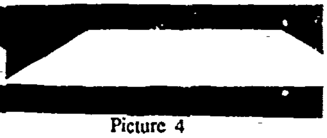
Picture 1



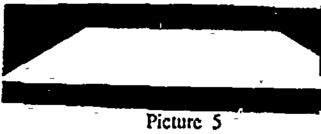
Picture 2



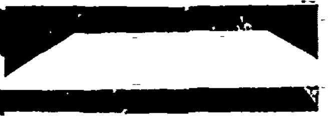
Picture 3



Picture 4

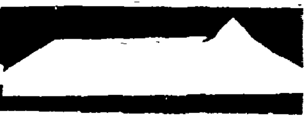


Picture 5



Picture 6

Picture 7



Picture 8

REFERENCE

1) Jeffrey M. Augenbaum. The Pseudospectral Method for Limited - Area Elastic Wave Calculations. February 13, 1990.

2) Bengt Fornberg. The Pseudospectral method Comparisions With Finite Differences for the Elastic Wave Equation. GEOPHYSICS. VOL. 52, NO. 4 (APRIL 1987), P. 483 501

3) A. R. Mitchell and D. F. Griffiths *The Finite Difference Method in Partial Differential Equations* John Wiley and Sons, 1980.

MODELLING AND SIMULATION APPROACH TO STRUCTURAL REARRANGEMENT OF DISTILLATION COLUMN

M.Atanasijević, R.Karba, B.Zupančič, F.Bremšak
Faculty of Electrical and Computer Engineering
Tržaška 25, 61000 Ljubljana, Yugoslavia

ABSTRACT - The paper deals with mathematical modelling of industrial continous distillation column for separation of four - component mixture composed of furfural, methanol, acetic acid and water. The aim of the work is optimisation of existing device in the sence of product quality improvement and minimisation of energy consumption. The model was examined by the aid of digital simulation language SIMCOS. Presented results show that the best solution lies in construction change which have good influence especially to the bottom product and so to the energy consumption of the device.

1. Plant description

Presently the input mixture is fed to the device at the 25th tray and furfural is taken out at the 28th tray as a side stream product. The second product is methanol which is separated in the upper part of the column. As it is shown in Figure 1, furfural mixture is kept in separating reservoir leaving the device with flow rate D_A. Furfural is heterogeneous azeotrope. In mentioned reservoir it separates to heavier component, which leaves the system with flow O and represents one of the final products while lighter component is taken to the bottom reservoir where it mixes with the industry environment flow. The energy needed for the operation of the column is supplied in the form of water vapor flow G at the bottom of the device.

2. Mathematical modelling and simulation of existing device

It is obvious that the most important variables of the system are input and output flow rates and their compositions. So we started modelling procedure by writing component and mass balances for each tray of the column (for the simplicity of notation the condenser is numbered as the 44th plate):

$$\begin{aligned}\frac{d(x_{ji}(t)m_{ji}(t))}{dt} = {} & L_{j+1}(t)x_{j+1,i}(t) - L_j(t)x_{ji}(t) + \\ & V_{j-1}(t)y_{j-1,i}(t) - V_j(t)y_{ji}(t) + \\ & L_{si,j}(t)x_{si,ji}(t) - L_{so,j}(t)x_{so,ji}(t) + \\ & V_{si,j}(t)y_{si,ji}(t) - V_{so,j}(t)y_{so,ji}(t)\end{aligned}$$

$$j = 1, \dots ,44; \qquad i = 1, \dots ,4;$$

where $L_j(t)$ and $V_j(t)$ denotes the liquid and vapor flow rates from the j-th tray, $x_{ji}(t)$ the liquid composition of i-th component on j-th tray and $y_{ji}(t)$ the vapor composition of i-th component upon j th tray, $L_{si,j}(t)$, $L_{so,j}(t)$, $V_{si,j}(t)$ and $V_{so,j}(t)$ are input and output liquid and vapor flow rate. on j-th tray with compositions $x_{si,ji}(t)$, $x_{so,ji}(t)$, $y_{si,ji}(t)$ and $y_{so,ji}(t)$.

A lot of methods can be used to describe the relationship between liquid and vapor compositions of mixture at defined pressure and temperature. Especially in the absence of reliable vapor-liquid equilibrium data the following equations are very popular in chemical engineering:

$$y_{ji}(t) = K_{ji}\, x_{ji}(t); \quad j = 1, \dots ,43; \quad i = 1, \dots ,4$$

where K_{ji} is distribution coefficient of i-th component at j-th tray of the column.

The analysis of measured data and simulation results, which we obtained by the aid of digital simulation language SIMCOS, gave the following conclusions:

- Concentration of methanol in the flow rate D_M is higher than the prescribed value so the G could be reduced.
- Concentration of furfural in the flow rate D_A is inside the prescribed values.
- Concentration of furfural in bottom flow B is too high thus demanding the vapor flow rate G to be enlarged for 10 to 15%. (Normal values of furfural concentration in bottom product were approximately 0.17%. Due to more rigorous ecological demands this values have to be reduced to 0.05 -0.1%. Simulation results and experiments on the plant have shown that for such improvement vapor flow rate G should be enlarged for 10 to 15%.)

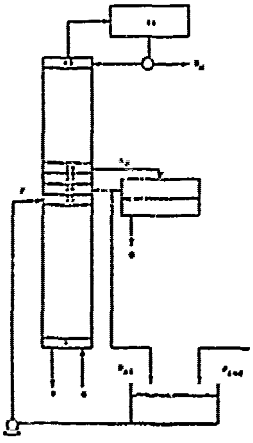


Figure 1. Schematic representation of discussed distillation column.

These conclusions indicate that vapor flow rate G should be enlarged because of ecological demands meaning that energy consumption will of course be also greater. Nothing really effective could be therefore made with this column structure.

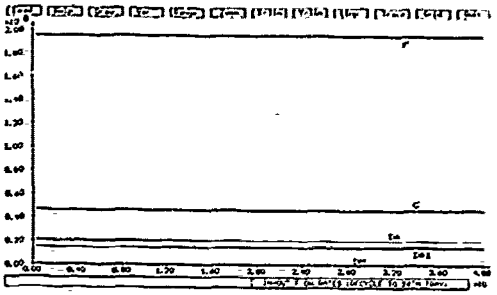


Figure 2. Input and output flow rates of the device.

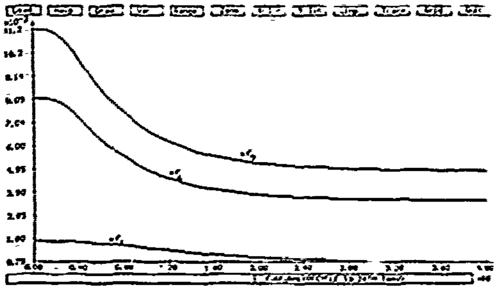


Figure 3.a. Concentration of furfural at the 1st, 6th and 7th tray.

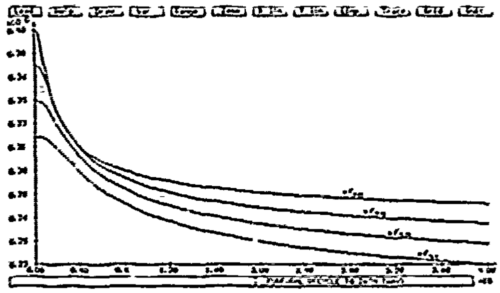


Figure 3.b. Concentration of furfural at the 28th to 31th tray.

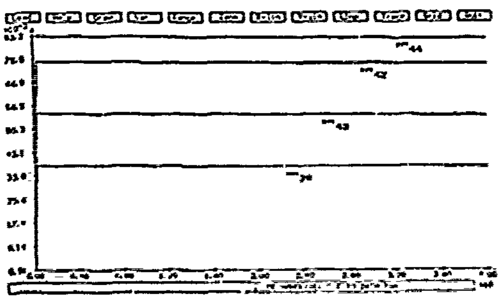


Figure 4. Concentration of methanol at the top of the column.

3. Structural rearrangement of distillation column

In the second step of modelling we decided to examine some possibilities in structural rearrangement of the device. The best result in this phase was given by the solution in Figure 1. The column is now fed back with the lighter furfural component from the separation reservoir on the 26th tray. This of course changed mass balances of 26th tray and reservoir in which ϕ_{ind} is entering. The concequence is, that also feed flow rate F and its concentration is changed. The steady state values of this changes can be evaluated from previous equations.

The simulation was made for changed working conditios of the plant and results are shown in Figures 2 to 4 for input and output folw rates, furfural and methanol. From this we can see that the steady state composition of all trays in the column are changed and that this rearrangement caused the lowering of concentrations of furfural in the bottom flow B to the prescribed values, leaving the other requirements fulfilled.

4. Conclusion

Experiments on the plant showed that the proposed assumptions were justified. We also assume that the model could benefit with the introduction of nonconstant distribution coefficients, what will be needed in later work for the improvement of the existing control loops. Digital simulation language SIMCOS we used was very efficient tool for solving complex mathematical models with a large number of variables, nonlinearities and measured data and saved a lot of time and work on the real device.

References:

[1] Atanasijević M., R.Karba, F.Bremšak: Semibatch Distillation Modelling and Control Design, Proceedings of the 3th Symposium Simulationstechnik, Bad Munster a. St.-Ebernburg, Germany, pp. 464-468, 1985.

[2] Atanasijević M., R.Karba, F.Bremšak, T.Recelj, J.Golob, L.Fele: Comparison of Three Different Approaches to the Semibatch Distillation Column Control Design, Preprints of IFAC International Symposium on Dynamic and Control of Chemical Reactors and Distillation Columns, Bournemouth, UK, pp. 231-236, 1986.

[3] Zupančič,B., D.Matko, R.Karba, M.Šega: SIMCOS - Digital Simulation Language with Hybrid Capabilities, Proceedings of the 4th Symposium Simulationstechnik, Zurich, Schwitzerland, pp. 205-212, 1987.

NONLINEAR FINITE ELEMENT ANALYSIS OF SLOW CRACK GROWTH

XIANGQIAO YAN, SHANYI DU and DUO WANG
Harbin Institute of Technology, Harbin, 150006, P.R.China

ABSTRACT—In this paper, a material-nonlinear finite element program of the quasi-three dimension problem developed by authors of this paper is described from a few respects. Further, it is illustrated how this program is used to analyze plane problems. Finally, a finite element analysis of the process of slow crack growth is made for a center-cracked specimen subjected to monotonically increased load until the point of fast fracture is reached. Numerical results and experimental results are compared.Variation laws of some fracture mechanics parameters are given with the stable crack growth.

I . INTRODUCTION

In a cracked body with toughened materials or in the plane-stress state, the crack will grow with monoton: ally and slowly increased load before the fast tracture is reached.Many scholars attracted their attention to it.They have done a lot for it[1-7].

Experiments[8,9] showed that there is the stable delamination crack growth in some stacking sequence composite laminates. Mahishi[10] ever pointed out that while considerable progress has been made in understanding the delamination mechanisms in composites incorporating brittle or quasi-brittle matrix materials,the problem of toughened polyer matrix or metal matrix composite, in which large scale yielding is associated with the cracking, requires special attention.

In view of these, a material-nonlinear finite element program of the quasi-three dimension problem was recently developed by the authors [11]. Using this program, not only the nonlinear stress analyses, the delamination onset and the stable delamination crack growth for composite laminates but the nonlinear finite element analysis of stable crack growth process for plane cracked body are made. Here, the program is described from a few respects and a finite element analysis for a center-cracked specimen subjected to monotonically and slowly increased load until the point of fracture is reached is taken for example to check it. Finally, the variation laws of some fracture mechanics parameters are given with the stable crack growth.

II . DESCRIPTIONS OF PROGRAM

Here, a material nonlinear finite-element program [11] of the quasi three dimension problem developed by the authors will be described from a few respects.

A. GEOMETRICAL DESCRIPTION

The origin of the quasi-three dimension problem can refer to the Ref[13]. Its schamatic illustration and simplified model are shown in Fig.1.Its displacement field expressions are listed as follows[13]:

$$\begin{aligned} u &= e_0 x + U(y,z) \\ v &= V(y,z) \qquad\qquad (2\text{-}1) \\ w &= W(y,z) \end{aligned}$$

where e_0 is uniform action strain in the direction of x.

Strain-displacement relations are

$$\begin{aligned} &\varepsilon_x = u_{,x}\ ,\quad \varepsilon_y = v_{,y}\ ,\quad \varepsilon_z = w_{,z} \qquad (2\text{-}2) \\ &\gamma_{yz} = v_{,z} + w_{,y},\quad \gamma_{zx} = w_{,x} + u_{,z}\ ,\quad \gamma_{xy} = u_{,y} + v_{,x} \end{aligned}$$

It can be seen that the strains corresponding to displacement expressions (2-1) can be classfied into two parts,one part not depending on the coordinates, the other depending on the coordtnates, i.e.

$$\begin{aligned} [\bar{\varepsilon}]_0 &= [e_0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \qquad (2\text{-}3) \\ [\bar{\varepsilon}]_1 &= [0 \quad \varepsilon_y \quad \varepsilon_z \quad \gamma_{yz} \quad \gamma_{zx} \quad \gamma_{xy}] \end{aligned}$$

and that total strains are

$$[\bar{\varepsilon}] = [\bar{\varepsilon}]_0 + [\bar{\varepsilon}]_1$$

According to the displacement finite element method, we have

$$\begin{aligned} U &= \Sigma\ N_i U_i \qquad (2\text{-}4) \\ V &= \Sigma\ N_i V_i \\ W &= \Sigma\ N_i W_i \end{aligned}$$

where N_i are interpolation functions[14], U_i, V_i and W_i nodal displacements

Introducing the symbols:

$$\bar{\varepsilon}_i^* = \bar{\varepsilon}_{i+1} \quad (i=1,2,\cdots,5)$$

then the following results can be obtained from formulas (2-2) and (2-4):

$$[\bar{\varepsilon}]^F = [B][\delta]_e \qquad (2\text{-}5)$$

where $[\delta]^e$ is the element displacement vector, [B], for a triangle element, is the (5∧9) strain matrix whose expression is deleted.

B. MATERIAL-NONLINEAR FINITE ELEMENT EQUATIONS

Here, it is supposed that the individual lamina material in composite laminates is homogeneous. orthotropic and after yielding,follows Hill's ortho tropic plasticity theory.referred to the Ref [12].

In a material coordinate system,incremental stress-strain relations can be written as [12]:

$$[d\sigma] = [C]_{ep}[d\varepsilon] \qquad (2\text{-}6)$$

By coordinate translation for (2 6), incremental stress-strain relations expressed in the structural coordinate system can be obtained as follows:

$$[d\bar{\sigma}]=[\bar{C}]_{ep}[d\bar{\varepsilon}] \qquad (2\text{-}7)$$

where

$$[d\bar{\sigma}]=[T]^{-1}[d\sigma]$$
$$[d\bar{\varepsilon}]=[T]^{-1}[d\varepsilon]$$
$$[\bar{C}]_{ep}=[T]^{T}[C]_{ep}[T]$$

and [T] is the translation matrix whose expression is deleted here.

Introducing the following symbols:

$$\left.\begin{array}{l} d\bar{\sigma}_i^*=d\bar{\sigma}_{i+1} \\ d\bar{\varepsilon}_i^*=d\bar{\varepsilon}_{i+1} \end{array}\right\} \quad (i=1,2,\cdots,5) \qquad (2\text{-}8)$$

a series of derivations are made to find

$$[d\bar{\sigma}]^*=[d\bar{\sigma}]_o^*+[d\bar{\sigma}]_1^*$$
$$=[\bar{C}_{12}\ \bar{C}_{13}\ \bar{C}_{14}\ \bar{C}_{15}\ \bar{C}_{16}]^T_{ep}de_o+[\bar{D}]_{ep}[d\bar{\varepsilon}]^* \qquad (2\text{-}9)$$

in which

$$(\bar{D}_{mn})_{ep}=(\bar{C}_{m+1\ n+1})_{ep} \quad (m,n=1,2,\cdots,5) \qquad (2\text{-}10)$$

and de_o expresses the increment of e_o.

It is noted that. in a material coordinate system, the elastic-plastic matrix $[C]_{ep}$ can be expressed in terms of the elastic $[C]_e$ and the plastic matrix $[C]_p$, i.e.

$$[C]_{ep}=[C]_e-[C]_p \qquad (2\text{-}11)$$

Similarly, in a structural coordinate system, the elastic-plastic matrix $[\bar{C}]_{ep}$ can also be expressed in terms of the elastic matrix $[\bar{C}]_e$ and the plastic matrix $[\bar{C}]_p$, i.e.

$$[\bar{C}]_{ep}=[\bar{C}]_e-[\bar{C}]_p \qquad (2\text{-}12)$$

in which

$$[\bar{C}]_e=[T]^T[C]_e[T]$$
$$[\bar{C}]_p=[T]^T[C]_p[T]$$

Introducing the symbols:

$$\left.\begin{array}{l} (\bar{D}_{mn})_e=(\bar{C}_{m+1\ n+1})_e \\ (\bar{D}_{mn})_p=(\bar{C}_{m+1\ n+1})_p \end{array}\right\} \quad (m,n=1,2,\cdots,5) \qquad (2\text{-}13)$$

then the following results can be obtained from formulas (2-10) and (2-12)

$$[\bar{D}]_{ep}=[\bar{D}]_e-[\bar{D}]_p \qquad (2\text{-}14)$$

By substituting (2-14) into (2-9), one can find

$$[d\bar{\sigma}]^*=[d\bar{\sigma}]_o^{*'}+[\bar{D}]_e[d\bar{\varepsilon}]^* \qquad (2\text{-}15)$$

in which

$$[d\bar{\sigma}]_o^{*'}=[d\bar{\sigma}]_o^*-[\bar{D}]_p[d\bar{\varepsilon}]^* \qquad (2\text{-}16)$$

and

$$[d\bar{\sigma}]^*=[\bar{C}_{12}\ \bar{C}_{13}\ \bar{C}_{14}\ \bar{C}_{15}\ \bar{C}_{16}]^T_{ep}de_o \qquad (2\text{-}17)$$

According to the formulas (2-15), we know, if the $[d\bar{\sigma}]_o^{*'}$ are regarded as original stresses, then the material-nonlinear problem is translated into the material linear problem possessing the original stresses $[d\bar{\sigma}]_o^{*'}$[14]. It can be seen from formulas (2-16) and (2-17) that the original stresses $[d\bar{\sigma}]_o^{*'}$ depend on not only the stress state before immediate loading but the strains $[d\bar{\varepsilon}]^*$ due to immediate loading. Thus, the original stress vector [14] translated by original stresses $[d\bar{\sigma}]_o^*$ depends on the immediate strain increment $[d\bar{\varepsilon}]^*$. The following finite element equations can be derived by using stress-strain relations (2-15), strain-displacement relations (2-5) and the variation priciple:

$$[K][d\delta]_i=[dR]_i-[dR]_{oi}+[dR(\Delta\varepsilon)]_i \qquad (2\text{-}18)$$

where the subscript i expresses the ith step loading increment;

[K] the elastic structural stiffness matrix, in which an element stiffness matrix is

$$[K]_e=\int_v[B]^T[\bar{D}]_e[B]dv; \qquad (2\text{-}19)$$

$[d\delta]_i$ the displacement vector due to the ith step loading increment;

$[dR]_i$ the load vector due to the ith step loading increment;

$[dR]_{oi}$ the original stress vector translated by $[d\bar{\sigma}]_o^*$ in which an element original stress vector is

$$[dR]_{oei}=\int[B]^T[d\bar{\sigma}]_{oi}^*dv \qquad (2\text{-}20)$$

Because the original stress vector $[dR(\Delta\varepsilon)]_i$ depends on the immediate strain increment $[d\bar{\varepsilon}]^*$, and the immediate strain increment $[d\bar{\varepsilon}]^*$ is also unknown, equations (2-18) must be solved by using an iterative method[14]. After finding the approximate $[d\bar{\varepsilon}]^*$, the calculation formula of $[dR(\Delta\varepsilon)]_{ie}$ is

$$[dR(\Delta\varepsilon)]_{ie}=\int[B]^T[\bar{D}]_{pi}[d\varepsilon]_i^*dv \qquad (2\text{-}21)$$

C. LOADING AND UNLOADING CRITERIA

Since the local unloading around the crack tip due to stable crack growth occurs, loading and unloading problem are necessarily considered in simulating the stable crack growth.

The yielding equation for a homogenous, orthotropic material can be expressed in terms of the effective stress

$$2g(\sigma_{ij}\ \ h)=\frac{2}{3}(F_o+G_o+H_o)\bar{\sigma}^2-h^2=0 \qquad (2\text{-}22)$$

where what individual symbols in (2-22) mean can refer to the Ref [12].

Introducing the following symbols:

$$f(\sigma_{ij},\ h')=\frac{3}{F_o+G_o+H_o}\cdot g(\sigma_{ij},\ h) \qquad (2\text{-}23)$$

in which

$$h'=\frac{h^2}{\frac{2}{3}(F_o+G_o+H_o)}$$

then the yielding equation can be expressed as

$$f(\sigma_{ij},\ h')=\bar{\sigma}^2-h'^2=0 \qquad (2\text{-}24)$$

where the original value of h'.h'ₒ, is

$$h'_o=\sqrt{\frac{3}{2(F_o+G_o+H_o)}} \qquad (2\text{-}25)$$

The following relations for a homogeneous isotropic material exist:

$$3F_o=3G_o=3H_o=L_o=M_o=N_o \qquad (2\text{-}26)$$

and

$$F_o=1/(2\sigma_y^2)$$

where σ_y is the original yielding stress. and equation (2-25) can be translated into

$$h'_o=\sigma_y \qquad (2\text{-}27)$$

At this moment, the equation (2-24) is the original yielding equation for a homogeneous,isotropic material.

Generally, the total strain increments are classified into elastic strain increments and plastic strain increments. i.e.

$$[d\bar{\varepsilon}]=[d\bar{\varepsilon}]_e+[d\bar{\varepsilon}]_p \qquad (2\text{-}28)$$

According to the equation (2-24), the unloading criterion can be expressed as

$$f(\sigma_{ij}, h')<0 \quad \text{or } d\bar{\sigma}<0 \qquad (2\text{-}29)$$

At this moment. we have

$$[d\bar{\varepsilon}]_e^s=[\bar{S}][d\bar{\sigma}]^s \qquad (2\text{-}30)$$

and

$$[d\bar{\varepsilon}]_p^s=0 \qquad (2\text{-}31)$$

in which

$$[\bar{S}]=[\bar{D}]_e^{-1}$$

And the loading criterion can be expressed as

$$f(\sigma_{ij}, h')=0 \quad \text{or } d\bar{\sigma}>0 \qquad (2\text{-}32)$$

At this moment, elastic strain increments $[d\bar{\varepsilon}]_e^s$ are still determined by (2-30). while how plastic strain increments $[d\bar{\varepsilon}]_p^s$ are determined can be referred to the Ref [12].

D.ON TREATMENT OF PLANE PROBLEMS

A material-nonlinear finite element program of the quasi-three dimension problem developed by the authors can be used to analyze plane problems.

Displacement expressions for a plane strain problem are

$$\begin{aligned} u&=0 \\ v&=V(y,z) \qquad (2\text{-}33)\\ w&=W(y,z) \end{aligned}$$

It can be seen that if the limitations U(y.z)=0 and e_o=0 to the displacement expressions (2-1) are made. then displacement expressions (2-1) become the displacement expressions (2-33). Further. that the appropriate limitations to material constants, including the elastic constants and strength constants. of homogeneous, orthotropic continua. are made will cause these continua to become homogeneous.isotropic continua. And it is noted that if the constants in equilibrium equations. expressed by displacements.of a plane strain problem, are substituted by new appropriate constants. these equations will become the equilibrium equations of a plane stress problem. Thus, a quasi-three dimension finite element program can be used to analyze not only a quasi-three-dimension problem,if the appropriate limitations to boundary displacement conditions of a quasi-three dimension problem are made. but also a plane problem.

Ⅲ. NUMERICAL SIMULATION OF STABLE PLANE CRACK GROWTH

Here. the material-nonlinear finite element program developed by the authors will be used to simulate the stable crack growth process of a plane crack[5]. A rectangular plate. made of 2024-T3 aluminum alloy. with a centered line crack,is subjected to monotonically and slowly increased loading up to the onset of fast fracture. The geometric configuration and the loading condition of the center-cracked specimen is shown in Fig.2.

The geometrical parameters. i.e.. length 2l. width 2W,thickness B. initial crack size 2aₒ are listed as follows:

$$\begin{aligned} &2L=27\text{in.} \quad 2W=12\text{in..} \\ &B=0.062\text{in..}\ 2a_o=6\text{in.} \end{aligned} \qquad (3\text{-}1)$$

The material property parameters,i.e..Young's modulus. poission's ratio. yield stress and linear hardening modulus are

$$\begin{aligned} &E=10324\text{Ksi}, \quad \nu=0.33. \\ &\sigma_y=54.20\text{Ksi}.\ H'=259.74\text{Ksi} \end{aligned} \qquad (3\text{-}2)$$

The finite element mesh and two paths for the evaluation of J-integral are referred to the Ref[12].

Here. the load-crack size curve measured by experiment, as shown in Fig.3(a). is taken as the input data which is used the govening equation.The comparision of the experimental lead-displacement curve with the numerical load-displacement data. shown in Fig.3(b) proved that the numerical results here are correct.

A. J-INTEGRAL

Fig.4 showed that J-integral varied with crack size. It can be seen that the values of J-integral along the two paths are almost the same with the maximum difference being ±3.7% .This result was in agreement with that reported in [1.3]. Refs [1.3] pointed out that J-integral is still conservative in the condition of small amount of crack growth. Ref[1] reported that J-integral of far field for a compact tension specimen is in agreement with Jintegral of near field in the condition of $(a-a_o)/(W-a_o)\leqslant 0.06$. Here. $(a_c-a_o)/(W-a_o)=0.03$. Moreover. it can be seen that J-integral is proportional to the amount of crack growth in the origin of the stable crack growth. This result proved the two parameter charac terization of fracture toughness properties proposed by Shih.et al[1].

B. CRACK OPENING ANGLE

The two definitions of crack opening angle are. one is the average crack opening angle. denoted by COA or α_0, which is defined as the ratio of the crack opening displacement at the original crack tip to the total amount of crack growth that has occurred, while the other is the crack tip opening angle, denoted by CTOA or α_1, which is defined as the ratio between crack opening displacement at a short and fixed distance Δ behind the current crack tip and Δ. In this work, Δ is taken to be the spacing between nodes located along the path of crack growth. The results of COA and CTOA are plotted in Fig.5. It can be seen that COA and CTOA are both varied with the stable crack growth, but the range of the variation of CTOA is smaller than that of COA. From numerical values, the data of COA snd CTOA reported here were almost the same as those reported in Ref [7].

C. CRACK TIP FORCE

The technique of releasing the crack tip forces is often used in the numerical simulation of stable crack growth. The result of crack tip force F_c is shown in Fig.6. It can be seen that F_c is basically constant with the stable crack growth. This result proved the mixed criterion of J_{c1} and F_c proposed by Kannunen, et,al[2].

Ⅳ. CONCLUSION

In this paper, a material-nonlinear finite element program of the quasi-three dimension problem developed by the authors is described from a few respects. Further,it is illustrated how this program is used to analyze plane problems. Finally, a finite element analysis of the process of slow crack growth is made for a center-cracked specimen subjected to monotonically increased load until the point of fast fracture is reached. Numerical results and experimental results are compared. Variation laws of some fracture mechanics parameters are given with the stable crack growth.

REFERENCES

[1] C.F.Shih,et.al., Elastic-Plastic Fracture, ASTM STP 668,1978,65-120.

[2] M.F.Kannine, et.al., Elastic-Plastic Fracture, ASTM-STP 668, 1978,121-150.

[3] J.W.Hutchinson, et.al., Elastic-Plastic Fracture, ASTM STP 668, 1978,37-64.

[4] J.D.Lee and H.Liebowitz,Computers and structures, Vol.8, 1978,p.403.

[5] Shanyi Du and J.D.Lee,Engineering Fracture Mechanics, Vol.16, No.2, 1982,229-245.

[6] Shanyi Du and J.D.Lee,Engineering Fracture mechanics, Vol.17,1983,p.172.

[7] Lin Dang, et.al., Transactions of Solid Mechanics (Chinese), No.2, 1987,p.169.

[8] E.F.Rybicki. et.al., J.Composite Meterials. Vol.11, 1977, 470-487.

[9] A.S.D.Wang, et.al., Report No. NADC-79056-60, Drexel University. Philadelphia, PA. 1982.

[10] D.F.Adams and J.Mahishi. NASA-CR-172598,1985.

[11] Xiangqiao,Yan. et.al.. Computers and Structures. Vol.36. 1990,1135-1139.

[12] Xiangqiao Yan, Ph.D.Thesis, Harbin Institute of Technology, Harbin. China. 1989.

[13] R.M.Jones. Mechanics of Composite Materials. Scripta Book Company, Washington. D.C., 1975.

[14] Daqiao Lee. the Complementary Book of Finite Element Methods (Chinese). Science Publishing House, 1979.

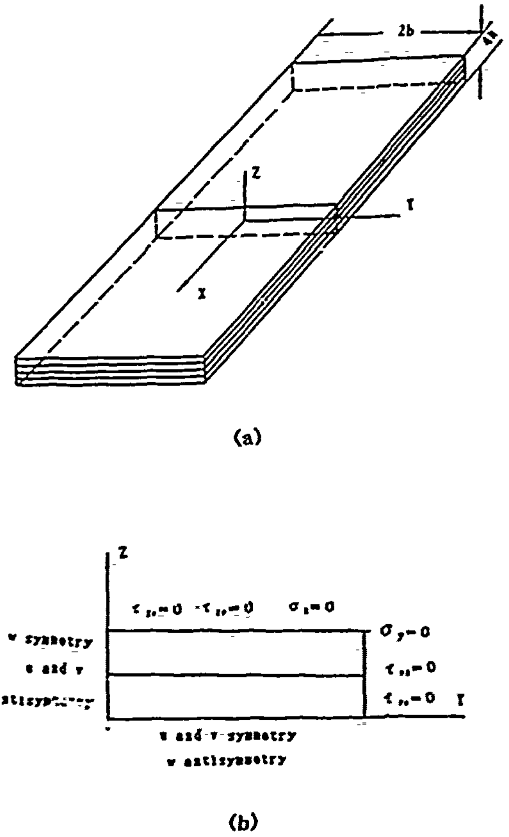


Fig.1 Schematic illustration of geometrical configuration (a) and simplified mode (b) of th- quasi-three dimension problem

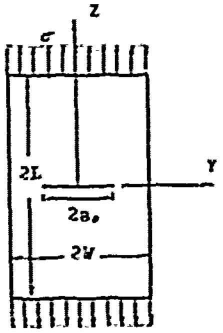


Fig.2 Geometrical configuration and loading conditions of a center-cracked plate

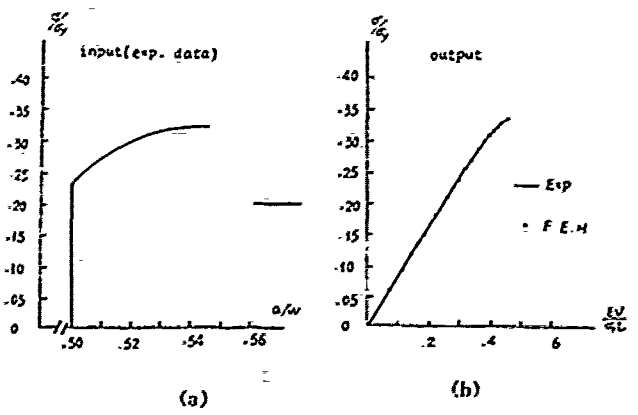


Fig.3 Comparision of experimental load-displacement curve with numerical load-displacement data

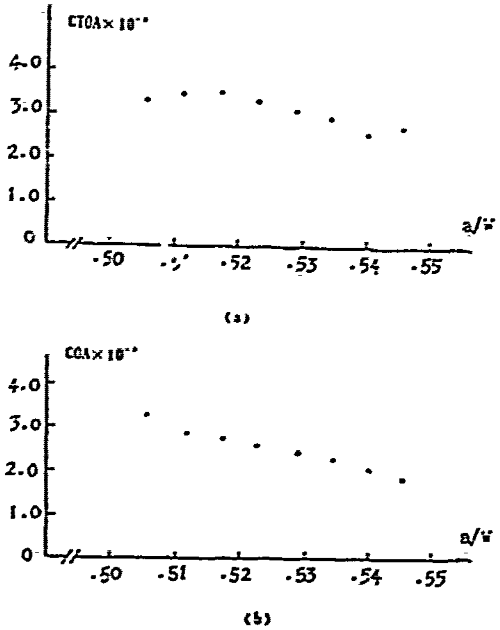


Fig.5 The variations of CTOA and COA with the stable crack growth

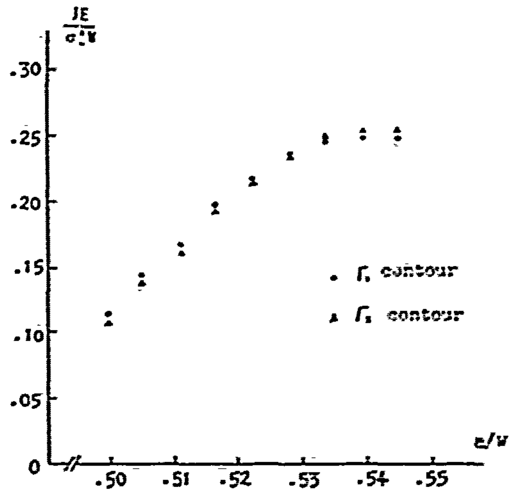


Fig.4 The comparision of J-integral along different integral paths

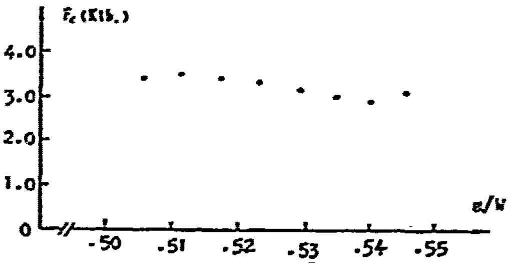


Fig.6 The variation of the crack tip node force Fc with the stable crack growth

HAMILTON'S PRINCIPLE AND THE EQUATIONS OF MOTION OF AN ELASTIC SHELL WITH AND WITHOUT FLUID LOADING[1]

Cleon E. Dean and Michael F. Werby
Naval Oceanographic and Atmospheric Research Laboratory
CODE 221, Building 1100
Stennis Space Center, MS 39529-5004

Abstract—It has proven quite difficult to employ exact elastodynamic theory to describe the behavior of elastic vibrations on arbitrary bounded shells. In addition, exact theories preclude direct interpretation of particular features observed due to the excitation o^c elastic shell surfaces. A rather interesting approach to describe surface vibrations may be obtained by constructing a Hamiltonian in some approximate form that assumes some correlation of motion of the outer and inner shell surface. The class of theories that allow for this approach are referred to in applied mechanics as shell theories. The interesting feature of this Hamiltonian approach is that one can add various physical mechanisms to the Hamiltonian such as extensional motion, rotary inertia, shear distortion, fluid loading, etc., and thereby study the individual contributions to resonance patterns while adding physical insight to the fundamental processes that occur on shell surfaces. We develop shell theories in this manner and examine various contributions via Hamilton's principle. We believe that fluid loading has by and large not been treated adequately in the past, and we place particular emphasis on the treatment of that contribution to this work.

I. INTRODUCTION

The usual assumptions in shell theory (due to A. E. H. Love[2]) are: (1) The thickness of a shell is small compared with the smallest radius of curvature of the shell; (2) The displacement is small in comparison with the shell thickness; (3) The transverse normal stress acting on planes parallel to the shell middle surface is negligible; (4) Fibers of the shell normal to the middle surface remain so after deformation and are themselves not subject to elongation.

We shall use these assumptions in the development of a shell theory in the style of a Timoshenko-Mindlin type plate theory.

II. DERIVATION OF EQUATIONS OF MOTION

For spherical shells, membrane stresses (proportional to β) predominate over flexural stresses (proportional to β^2) where

$$\beta = \frac{1}{\sqrt{12}}\frac{h}{a}.$$

We differ from the standard derivation for the sphere (due to Junger and Feit[3]) by retaining all terms of order β^2 in both the kinetic and potential energy parts of the Lagrangian. We note that this level approximation will allow us to include the effects of rotary inertia and shear distortion in our shell theory, as well as the usual extensional motion of the shell. The new derivation is as follows: let a u, v, w axis system be set up on a spherical shell of radius a (measured to mid-shell) with thickness h, as shown in Fig. 1.

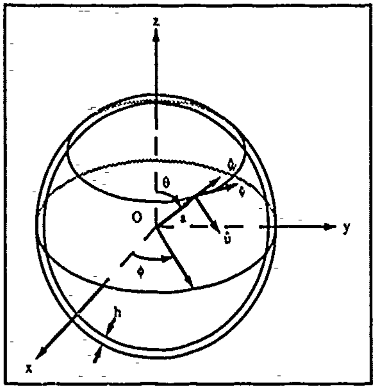


Fig. 1. Spherical shell showing coordinate systems used.

Then the new Lagrangian (which is equivalent to a Timoshenko-Mindlin theory as applied to a spherical shell) is $L = T - V + W$, where the kinetic energy is

$$T = \frac{1}{2}\rho_s \int_0^{2\pi}\int_0^{\pi}\int_{-h/2}^{h/2} (\dot{u}_s^2 + \dot{w}_s^2)(a+x)^2 \sin\theta\, dx\, d\theta\, d\phi, \tag{1}$$

with the surface displacements assumed linear in Timoshenko-Mindlin fashion:

$$\dot{u}_s = (1+\frac{x}{a})\dot{u} - \frac{x}{a}\frac{\partial \dot{w}}{\partial\theta}, \quad \dot{w}_s = \dot{w}. \tag{2a, b}$$

There is no movement in the v-direction since the ensonifying field can be taken to be torsionless. By substitution, the kinetic energy becomes

$$T = \pi\rho_s \int_0^{\pi} \sin\theta[(\frac{h^5}{80a^2} + \frac{h^3}{2} + ha^2) - 2(\frac{h^5}{80a^2} + \frac{h^3}{4})\dot{u}\frac{\partial \dot{w}}{\partial\theta}$$
$$+(\frac{h^5}{80a^2} + \frac{h^3}{12})(\frac{\partial \dot{w}}{\partial\theta})^2 + (\frac{h^3}{12} + ha^2)\dot{w}^2]d\theta, \tag{3}$$

or finally,

$$T = \pi\rho_s ha^2 \int_0^{\pi}[(1.8\beta^4 + 6\beta^2 + 1)\dot{u}^2 - (3.6\beta^4 + 6\beta^2)\dot{u}\frac{\partial \dot{w}}{\partial\theta}$$
$$+(1.8\beta^4 + \beta^2)(\frac{\partial \dot{w}}{\partial\theta})^2 + (\beta^2 + 1)\dot{w}^2]\sin\theta d\theta. \tag{4}$$

which to order β^2 is

$$T \approx \pi\rho_s ha^2 \int_0^{\pi}[(1 + 6\beta^2)\dot{u}^2 - 6\beta^2\dot{u}\frac{\partial \dot{w}}{\partial\theta} + \beta^2(\frac{\partial \dot{w}}{\partial\theta})^2$$
$$+(1+\beta^2)\dot{w}^2]\sin\theta d\theta. \tag{5}$$

In a similar fashion the potential energy is

$$V = \frac{1}{2}\int_0^{\pi}\int_0^{2\pi}\int_{-h/2}^{h/2}(\sigma_{\theta\theta}\varepsilon_{\theta\theta} + \sigma_{\phi\phi}\varepsilon_{\phi\phi})(x+a)^2 \sin\theta dx d\theta d\phi, \tag{6a}$$

$$= \frac{1}{2}\int_0^{\pi}\int_0^{2\pi}\int_{-h/2}^{h/2}[\frac{E}{1-v^2}\frac{1}{(x+a)^2}([(1+\frac{x}{a})\frac{\partial u}{\partial\theta} - \frac{x}{a}\frac{\partial^2 w}{\partial\theta^2} + w]^2$$
$$+\{\cot\theta[(1+\frac{x}{a})u - \frac{x}{a}\frac{\partial w}{\partial\theta}] + w\}[(1+\frac{x}{a})\frac{\partial u}{\partial\theta} - \frac{x}{a}\frac{\partial^2 w}{\partial\theta^2} + w])]$$
$$(x+a)^2 \sin\theta dx d\theta d\phi. \tag{6b}$$

$$= \frac{\pi Eh}{1-v^2}\int_0^{\pi}\{(w + \frac{\partial u}{\partial\theta})^2 + (w + u\cot\theta)^2$$
$$+2v(w + \frac{\partial u}{\partial\theta})(w + u\cot\theta) + \beta^2[(\frac{\partial u}{\partial\theta} - \frac{\partial^2 w}{\partial\theta^2})^2$$
$$+\cot^2\theta(u - \frac{\partial w}{\partial\theta})^2 + 2v\cot\theta(u - \frac{\partial w}{\partial\theta})(\frac{\partial u}{\partial\theta} - \frac{\partial^2 w}{\partial\theta^2})]\}\sin\theta d\theta, \tag{6c}$$

[1] We wish to thank the Office of Naval Research, the Office of Naval Technology, and NOARL Management including Drs. Chin-Bing, Franchi, and Moseley for support of this work. Dr. Dean is at NOARL on an ONT Fellowship. This work was funded by NOARL Program Element 61153N, H. Morris, Program Manager. NOARL contribution PR91:068:221.

[2] A. E. H. Love, *A Treatise on the Mathematical Theory of Elasticity* (New York: Dover, 1944), Ch. XXIV.

[3] M. C. Junger and D. Feit, *Sound, Structures, and Their Interaction*, 2nd ed. (Cambridge, MA: MIT Press, 1986), p. 228 ff.

where the nonvanishing strain components are

$$\varepsilon_{\theta\theta} = \frac{1}{a}\left(\frac{\partial u}{\partial\theta} + w\right) + \frac{x}{a^2}\left(\frac{\partial u}{\partial\theta} - \frac{\partial^2 w}{\partial\theta^2}\right), \tag{7a}$$

and

$$\varepsilon_{\phi\phi} = \frac{1}{a}(\cot\theta u + w) + \frac{x}{a^2}\cot\theta\left(u - \frac{\partial w}{\partial\theta}\right). \tag{7b}$$

The nonzero stress components are

$$\sigma_{\theta\theta} = \frac{E}{1-\nu^2}(\varepsilon_{\theta\theta} + \nu\varepsilon_{\phi\phi}), \quad \sigma_{\phi\phi} = \frac{E}{1-\nu^2}(\varepsilon_{\phi\phi} + \nu\varepsilon_{\theta\theta}). \tag{8 a,b}$$

The work done by the surrounding fluid on the sphere is

$$W = 2\pi a^2 \int_0^\pi p_a w \sin\theta d\theta. \tag{9}$$

Since the integration along the polar angle is intrinsic to the problem, the solution must be found using a *Lagrangian density*:

$$\begin{aligned} \mathrm{L}_1 = {} & \pi\rho_s h a^2[(1+6\beta^2)\dot{u}^2 - 6\beta^2\dot{u}\frac{\partial\dot{w}}{\partial\theta} + \beta^2(\frac{\partial\dot{w}}{\partial\theta})^2 \\ & +(1+\beta^2)\dot{w}^2]\sin\theta - \frac{\pi Eh}{1-\nu^2}\{(w+\frac{\partial u}{\partial\theta})^2 + (w+u\cot\theta)^2 \\ & +2\nu(w+\frac{\partial u}{\partial\theta})(w+u\cot\theta) + \beta^2[(\frac{\partial u}{\partial\theta} - \frac{\partial^2 w}{\partial\theta^2})^2 \\ & +\cot^2\theta(u-\frac{\partial w}{\partial\theta})^2 + 2\nu\cot\theta(u-\frac{\partial w}{\partial\theta})(\frac{\partial u}{\partial\theta} - \frac{\partial^2 w}{\partial\theta^2})]\}\sin\theta \\ & +2\pi a^2 p_a w\sin\theta, \end{aligned} \tag{10}$$

with differential equations

$$\frac{\partial \mathrm{L}_1}{\partial u} - \frac{d}{d\theta}\frac{\partial \mathrm{L}_1}{\partial u_\theta} - \frac{d}{dt}\frac{\partial \mathrm{L}_1}{\partial u_t} = 0, \tag{11a}$$

and

$$\frac{\partial \mathrm{L}_1}{\partial w} - \frac{d}{d\theta}\frac{\partial \mathrm{L}_1}{\partial w_\theta} - \frac{d}{dt}\frac{\partial \mathrm{L}_1}{\partial w_t} = 0. \tag{11b}$$

By substitution we have

$$\begin{aligned} & (1+\beta^2)\left[\frac{\partial^2 u}{\partial\theta^2} + \cot\theta\frac{\partial u}{\partial\theta} - (\nu+\cot^2\theta)u\right] - \beta^2\frac{\partial^3 w}{\partial\theta^3} \\ & -\beta^2\cot\theta\frac{\partial^2 w}{\partial\theta^2} + [(1+\nu)+\beta^2(\nu+\cot^2\theta)]\frac{\partial w}{\partial\theta} \\ & -\frac{a^2}{c_p^2}[(1+6\beta^2)\ddot{u} - 3\beta^2\frac{\partial\ddot{w}}{\partial\theta}] = 0, \end{aligned} \tag{12a}$$

and

$$\begin{aligned} & \beta^2\frac{\partial^3 u}{\partial\theta^3} + 2\beta^2\cot\theta\frac{\partial^2 u}{\partial\theta^2} - [(1+\nu)(1+\beta^2)+\beta^2\cot^2\theta)]\frac{\partial u}{\partial\theta} \\ & +\cot\theta[(2-\nu+\cot^2\theta)\beta^2 - (1+\nu)]u - \beta^2\frac{\partial^4 w}{\partial\theta^4} - 2\beta^2\cot\theta\frac{\partial^3 w}{\partial\theta^3} \\ & +\beta^2(1+\nu+\cot^2\theta)\frac{\partial^2 w}{\partial\theta^2} - \beta^2\cot\theta(2-\nu+\cot^2\theta)\frac{\partial w}{\partial\theta} \\ & -2(1+\nu)w - \frac{a^2}{c_p^2}(1+\beta^2)\ddot{w} = -p_a\frac{(1-\nu^2)a^2}{Eh}. \end{aligned} \tag{12b}$$

These differential equations have solutions of the form

$$u(\eta) = \sum_{n=0}^{\infty} U_n(1-\eta^2)^{1/2}\frac{dP_n}{d\eta}, \quad w(\eta) = \sum_{n=0}^{\infty} W_n P_n(\eta), \tag{13a,b}$$

where $\eta = \cos\theta$ and $P_n(\eta)$ are the Legendre polynomials of the first kind of order n. The differential equations of motion (12a,b) are satisfied if the expansion coefficients U_n and W_n satisfy a homogeneous system of linear equations whose determinant yields a frequency equation of the form

$$\begin{aligned} 0 = {} & \Omega^4(1+f_n) - \Omega^2[(1+\beta^2)\kappa(1+f_n) + 2(1+\nu) + \beta^2\kappa\lambda_n] \\ & +[2(1+\nu)+\beta^2\kappa\lambda_n](1+\beta^2)\kappa - [\beta^2\kappa + (1+\nu)]^2\lambda_n, \end{aligned} \tag{14}$$

where $\kappa = \nu + \lambda_n - 1$ and $\lambda_n = n(n+1)$. The fluid loading is including by the term f_n, which uses the exact volume equation of a sphere to include the influence of the exterior spherical shell of fluid that influences the motion of the vibrating spherical shell. This is done in a manner analogous to that for the fluid loaded plate: for a plate, fluid loaded on one side, of mass per unit area $\rho_s h$, the appropriate nondimensional measure of fluid-loading at any frequency ω is $\rho c/\omega\rho_s h$. The surface pressure for a sphere in terms of modal specific acoustic impedances z_n is

$$p(a,\theta,\phi) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} z_n \dot{W}_{mn} P_n^m(\cos\theta)\cos m\phi, \tag{15}$$

where

$$z_n \equiv i\rho c\frac{h_n(ka)}{h_n'(ka)}. \tag{16a}$$

Let

$$z_n \equiv r_n - i\omega m_n. \tag{16b}$$

Then

$$r_n = \rho c \mathrm{Re}\left[\frac{ih_n(ka)}{h_n'(ka)}\right]. \tag{17a}$$

Similarly

$$m_n = \rho c \mathrm{Im}\left[\frac{ih_n(ka)}{h_n'(ka)}\right]. \tag{17b}$$

The fluctuating pressure on the surface of the sphere constitutes the *radiation loading*. In Junger and Feit's derivation of the fluid loading,

$$f_n = \frac{m_n}{\rho_s h}, \tag{18}$$

but we use

$$f_n = \frac{3a^3 m_n}{\rho_s[a^3 - (a-h)^3]}, \tag{19}$$

which has the advantage of an exact volume difference for the amount of fluid affecting the vibrating shell.

III. CONCLUSIONS

We expect this new derivation of the equations of motion for a spherical shell to give new insight into the types of waves supported by a fluid loaded spherical or spheroidal shell. From previous work[4] we expect the effects of fluid loading to be more important for antisymmetric than for symmetric modes of vibration, and for our improved volume measure to be more significant for thicker shells. We also hope to see an improvement in the asymptotic behavior of the resonances in the high size parameter limit for the antisymmetric case. That is, we expect the results to go to the Rayleigh velocity limit as the relative radius of the shell goes to infinity.

[4] C. E. Dean and M. F. Werby, "Comparison of backscattered echoes predicted from exact theory and from thin-shell theories," *J. Acoust. Soc. Suppl.* 88, S15, 1990.

ASYMPTOTIC DERIVATION OF AN EVOLUTION THERMOELASTIC MODEL FOR LINEAR RODS

J. M. VIAÑO
Departamento de Matemática Aplicada
Facultad de Matemáticas
Universidad de Santiago de Compostela
15706 Santiago SPAIN

AND

L. ALVAREZ VAZQUEZ
Departamento de Matemática Aplicada
ETSI Telecomunicaciones
Universidad de Vigo
36280 Vigo SPAIN

Abstract.- In this paper, a dynamic thermoelastic model for linear beams is obtained by asymptotic analysis. The model is obtained as the weak limit of a rescaled three-dimensional model after a change of variable to a fixed domain. As well as, a compatibility condition in order to obtain strong convergence is given.

I. SETTING OF THE PROBLEM

The basic ideas of the asymptotic expansion method introduced by BERMUDEZ - VIAÑO [2] and TRABUCHO - VIAÑO [8,9] are applied here in order to obtain the justification of an evolution thermoelastic model for linearized rods. We study the case of an elastic beam $\Omega^\epsilon = \omega^\epsilon \times (0,L)$, $\varepsilon^2 = \text{meas}(\omega^\epsilon)$, clamped at both ends $\Gamma_0^\epsilon = \omega^\epsilon \times \{0,L\}$ and kept at them to an uniform reference temperature T_0. The beam is supposed to be submitted to body forces $F^\epsilon(x^\epsilon)$ in Ω^ϵ and surface forces $G^\epsilon(x^\epsilon)$ on $\Gamma^\epsilon = \partial\omega^\epsilon \times (0,L)$ and to initial conditions in displacement, velocity and temperature $(\bar{U}^\epsilon, \bar{V}^\epsilon, \bar{\Theta}^\epsilon)$. We assume the beam Ω^ϵ made of an isotropic, homogeneous, elastic meterial of Saint Venant - Kirchhoff, with Young's modulus E, Poisson's ratio ν, thermal dilation coefficient α, heat conductivity coefficient k and specific heat coefficient β, independent of ε. We denote by ρ^ϵ the mass density and we assume that $\rho^\epsilon = \varepsilon^2\rho$ with ρ independent of ε ([3],[7]). Then, the dynamic problem corresponding to the thermoelastic behavior of the beam Ω^ϵ during the time interval [0,T] is governed by the classical evolutive system of equations posed as a function of the displacement field U^ϵ and the temperature increment field $\vartheta^\epsilon = \zeta^\epsilon - T_0$ ([5]) :

$$\rho^\epsilon U_i^{\epsilon\,''} - \partial_j^\epsilon \Sigma_{ij}^\epsilon = F_i^\epsilon \quad \text{in } \Omega^\epsilon \times (0,T) , \qquad (1)$$

$$\beta\Theta^{\epsilon\,'} = \frac{1}{T_0}\, \partial_j^\epsilon(k\, \partial_j^\epsilon \Theta^\epsilon) - \frac{E\,\alpha}{1-2\nu}\, e_{kk}^\epsilon(U^{\epsilon\,'}) \quad \text{in} \quad \Omega^\epsilon \times (0,T) , \qquad (2)$$

$$U^\epsilon = 0 \ , \ \Theta^\epsilon = 0 \qquad \text{on } \Gamma_0^\epsilon \times (0,T) , \qquad (3)$$

$$\Sigma_{i\alpha}^\epsilon n_\alpha = G_i^\epsilon \ , \quad k\,\partial_\alpha^\epsilon \Theta^\epsilon n_\alpha = 0 \qquad \text{on } \Gamma^\epsilon \times (0,T) , \qquad (4)$$

$$U^\epsilon(0) = \bar{U}^\epsilon \ , \ U^{\epsilon\,'}(0) = \bar{V}^\epsilon \ , \ \Theta^\epsilon(0) = \bar{\Theta}^\epsilon . \qquad (5)$$

In (1)-(5) the Piola - Kirchhoff tensor Σ^ϵ is given by the linear thermoelastic law :

$$e^\epsilon(U^\epsilon) - \alpha\, \Theta^\epsilon I = \frac{1+\nu}{E}\Sigma^\epsilon - \frac{\nu}{E}\, \text{tr}(\Sigma^\epsilon)\, I , \qquad (6)$$

where $e_{ij}^\epsilon(U^\epsilon) = \frac{1}{2}(\partial_i^\epsilon U_j^\epsilon + \partial_j^\epsilon U_i^\epsilon)$.

In a classical way the existence and uniqueness of solution of the three-dimensional problem (1)-(5) with the following regularity, is obtained ([6]) :

$$U^\epsilon \in C^0(0,T;(H^2(\Omega^\epsilon))^3) \cap C^1(0,T;\, (H^1(\Omega^\epsilon))^3) \cap$$
$$\cap\, C^2(0,T;(L^2(\Omega^\epsilon))^3) ,$$
$$\vartheta^\epsilon \in C^0(0,T;H^2(\Omega^\epsilon)) \cap C^1(0,T;L^2(\Omega^\epsilon)) .$$

II. WEAK CONVERGENCE

Following the works by BLANCHARD - FRANCFORT [3], BERMUDEZ - VIAÑO [2] and TRABUCHO - VIAÑO [8,9] we study the behavior of the three-dimensional model as the area of the cross-section of the beam tends to zero. In order to make this possible we consider a variational formulation of the problem. Then, we apply the following change of variable and scale, which allows us to work on the fixed domain $\Omega = \omega \times (0,L)$, $\omega = \varepsilon^{-1}\omega^\epsilon$:

$$\pi^\epsilon : x=(x_1, x_2, x_3) \in \Omega \to \pi^\epsilon(x) = x^\epsilon = (\varepsilon x_1 , \varepsilon x_2 , x_3) \in \Omega^\epsilon , \quad (7)$$

$$\begin{aligned}
&u_\alpha(\varepsilon) = \varepsilon\, U_\alpha^\epsilon \circ \pi^\epsilon \ , \quad u_3(\varepsilon) = U_3^\epsilon \circ \pi^\epsilon \ , \ \vartheta(\varepsilon) = \Theta^\epsilon \circ \pi^\epsilon , \\
&\sigma_{\alpha\beta}(\varepsilon) = \varepsilon^{-2}\,\Sigma_{\alpha\beta}^\epsilon \circ \pi^\epsilon, \ \sigma_{\alpha 3}(\varepsilon) = \varepsilon^{-1}\,\Sigma_{\alpha 3}^\epsilon \circ \pi^\epsilon, \ \sigma_{33}(\varepsilon) = \Sigma_{33}^\epsilon \circ \pi^\epsilon , \\
&f_\alpha(\varepsilon) = \varepsilon^{-1} F_\alpha^\epsilon \circ \pi^\epsilon \ , \qquad f_3(\varepsilon) = F_3^\epsilon \circ \pi^\epsilon \ , \qquad\qquad (8)\\
&g_\alpha(\varepsilon) = \varepsilon^{-2} G_\alpha^\epsilon \circ \pi^\epsilon \ , \qquad g_3(\varepsilon) = \varepsilon^{-1} G_3^\epsilon \circ \pi^\epsilon \ .
\end{aligned}$$

We denote $(u(\varepsilon), \bar{v}(\varepsilon), \bar{\vartheta}(\varepsilon))$ the element obtained from $(\bar{U}^\epsilon, \bar{V}^\epsilon, \bar{\Theta}^\epsilon)$ by the transformation (8) and $e_{ij}(v) = \frac{1}{2}(\partial_i v_j + \partial_j v_i)$ for $v \in (H^1(\Omega^\epsilon))^3$.

Theorem .- We assume the following weak convergences as ε tends to zero:

$$\begin{aligned}
&\bar{u}(\varepsilon) \to \bar{u}^0 \ \text{ weakly in } H = W^3; \quad W = (z \in H^1(\Omega^\epsilon) : z = 0 \text{ on } \Gamma_0) \\
&\bar{v}_\alpha(\varepsilon) \to \bar{v}_\alpha^0 \ , \quad \varepsilon\, \bar{v}_3(\varepsilon) \to \bar{v}_3^0 \ , \qquad \bar{\vartheta}(\varepsilon) \to \bar{\vartheta}^0 \ , \\
&\varepsilon^{-2} e_{\alpha\beta}(\bar{u}(\varepsilon)) \to \bar{e}_{\alpha\beta}^0 \ , \quad \varepsilon^{-1} e_{\alpha 3}(\bar{u}(\varepsilon)) \to \bar{e}_{\alpha 3}^0 \ , \\
&e_{33}(\bar{u}(\varepsilon)) \to \bar{e}_{33}^0 \quad \text{weakly in } L^2(\Omega) , \qquad\qquad (9)\\
&f_i(\varepsilon) \to f_i^0 \ , \ \partial_3 f_3(\varepsilon) \to \partial_3 f_3^0 \ \text{ weakly in } H^1(0,T;\, L^2(\Omega)) , \\
&g_i(\varepsilon) \to g_i^0 \ , \ \partial_3 g_3(\varepsilon) \to \partial_3 g_3^0 \ \text{ weakly in } H^1(0,T;\, L^2(\Gamma)) .
\end{aligned}$$

Then, for the sequence $(u(\varepsilon), \vartheta(\varepsilon))$ and, at least, for a subsequence of $(\sigma(\varepsilon), \nabla\vartheta(\varepsilon), e(u(\varepsilon)))$, noted in the same way, we have :

$u(\varepsilon) \to u^0$ weak-* in $L^\infty(0,T;H)$,

$u_\alpha(\varepsilon)' \to u_\alpha^{0\cdot}$, $\varepsilon\, u_3(\varepsilon)' \to 0$, $\varepsilon^2 \sigma_{\alpha\beta}(\varepsilon) \to \sigma_{\alpha\beta}^0$,

$\varepsilon\, \sigma_{\alpha 3}(\varepsilon) \to \sigma_{\alpha 3}^0$, $\sigma_{33}(\varepsilon) \to \sigma_{33}^0$,

$$\varepsilon^{-2} e_{\alpha\beta}(u(\varepsilon)) \to \frac{1+\nu}{E}\sigma_{\alpha\beta}^0 - \frac{\nu}{E}\sigma_{ii}^0\,\delta_{\alpha\beta} + \alpha\,\theta^0\,\delta_{\alpha\beta} \quad , \qquad (10)$$

$\varepsilon^{-1} e_{\alpha 3}(u(\varepsilon)) \to \frac{2(1+\nu)}{E}\sigma_{\alpha 3}^0$ weak-* in $L^\infty(0,T;L^2(\Omega))$,

$\theta(\varepsilon) \to \theta^0$ weak-* in $L^\infty(0,T;L^2(\Omega))$ and weakly in $L^2(0,T;$ W

$\varepsilon^{-1}\partial_\alpha\theta(\varepsilon) \to r_\alpha^0$ weakly in $L^2(0,T;L^2(\Omega))$,

where:

(i) u^0 is a Bernoulli-Navier field, $u^0 = (u_1^0, u_2^0, \underline{u}_3^0 - x_\alpha\,\partial_3 u_\alpha^0)$, whose components are determined by :

a) $u_\alpha^0 \in L^\infty(0,T;H_0^2(0,L))$ is the unique solution of the problem

$$\rho\, u_\alpha^{0\cdot\cdot} - E\, I_\alpha\, \partial_{3333} u_\alpha^0 = \int_\omega f_\alpha^0 + \int_\gamma g_\alpha^0 + \partial_3\Big(\int_\omega x_\alpha f_3^0 + \int_\gamma x_\alpha g_3^0\Big),$$

$$u_\alpha^0(0) = \bar{u}_\alpha^0 \quad , \quad u_\alpha^{0\cdot}(0) = \int_\omega \bar{v}_\alpha^0 \quad , \quad I_\alpha = \int_\omega (x_\alpha)^2 \ . \qquad (11)$$

b) $(\underline{u}_3^0, \theta^0) \in L^\infty(0,T;H_0^1(0,L)) \times L^\infty(0,T;H_0^1(0,L))$ is the unique solution of the coupled system :

$$E\,\partial_{33}\underline{u}_3^0 - E\,\alpha\,\partial_3\theta^0 = -\int_\omega f_3^0 - \int_\gamma g_3^0 \ ,$$

$$B\,\theta^{0\cdot} = \frac{k}{T_0}\partial_{33}\theta^0 - E\alpha\ \partial_3\underline{u}_3^{0\cdot} \ , \quad B = \beta + \frac{2E\alpha^2(1+\nu)}{1-2\nu} \ ,$$

$$B\,\theta^0(0) = \beta\int_\omega \bar{\theta}^0 + \frac{E\alpha}{1-2\nu}\int_\omega \bar{e}_{ii}^0 - E\alpha\int_\omega \partial_3\underline{u}_3^0(0) \qquad (12)$$

(ii) $\sigma_{\alpha i}^0$, r_α^0 are solutions, respectively, of :

$$\partial_\alpha\sigma_{\alpha i}^0 = 0 \ \text{ in } \omega \ , \quad \sigma_{\alpha i}^0\, n_\alpha = 0 \ \text{ on } \partial\omega \ . \qquad (13)$$

$$\partial_\alpha(k\, r_\alpha^0) = 0 \ \text{ in } \omega \ , \quad k\, r_\alpha^0\, n_\alpha = 0 \ \text{ on } \partial\omega \ . \qquad (14)$$

(iii) The stress tensor component σ_{33}^0 is given by :

$$\sigma_{33}^0 = E\,\partial_3\underline{u}_3^0 - E\,x_\alpha\,\partial_{33}u_\alpha^0 - E\,\alpha\,\theta^0 + \nu\,\sigma_{\alpha\alpha}^0 \ . \qquad (15)$$

As well as, the bending moments, normal stress and shear forces verify, respectively :

$$q_3(\varepsilon) = \int_\omega \sigma_{33}(\varepsilon) \to q_3^0 = E\,\partial_3\underline{u}_3^0 - E\,\alpha\,\theta^0 \ ,$$

$$m_\alpha(\varepsilon) = \int_\omega x_\alpha\,\sigma_{33}(\varepsilon) \to m_\alpha^0 = -E\,I_\alpha\,\partial_{33}u_\alpha^0 \ . \qquad (16)$$

$$q_\alpha(\varepsilon) = \int_\omega \sigma_{\alpha 3}(\varepsilon) \to q_\alpha^0 = \int_\omega x_\alpha f_3^0 + \int_\gamma x_\alpha g_3^0 + \partial_3 m_\alpha^0 \ . \qquad \blacksquare$$

We also obtain a result of regularity for the limit problem. Thus, under mild assumptions on the initial data and the applied loads, we obtain that u^0 and θ^0 satisfy:

$$u_\alpha^0 \in C^0(0,T;H_0^2(0,L)) \cap C^1(0,T;L^2(0,L)) \ ,$$

$$\underline{u}_3^0 \in C^0(0,T;H_0^1(0,L)) \ , \qquad \theta^0 \in C^0(0,T;L^2(0,L)) \ .$$

III. STRONG CONVERGENCE

Following the technique by RAOULT [7], based on the convergence of the norms in the Hilbert space $L^2(0,T,L^2(\Omega))$, we can prove that the convergences of (10) are strong if and only if a compatibility condition over initial data is satisfied (ALVAREZ [1]).

Finally, the same method can be applied, with slight modifications, to study of the behavior of the beam under different boundary conditions (beam clamped only at one end, heat flux over the surface of the beam...) and/or nonhomogeneous material.

This research was partially supported by European Project SCIENCE SCI*0473-C (EDB) and by Project PB87-0481 of DGICYT of Spain.

REFERENCES

[1] ALVAREZ VAZQUEZ, L. "Comportamiento asintotico de algunos problemas en elasticidad de placas y vigas". Thesis, Univ. Santiago de Compostela.(To appear in 1991)

[2] BERMUDEZ, A.- VIAÑO, J.M. "Une justification des equations de la thermoelasticite des poutres a section variable par des methodes asymptotiques." RAIRO Analyse Numerique, 18 (1984) , 347-376.

[3] BLANCHARD, D. - FRANCFORT, G. A. "Asymptotic thermo- elastic behavior of flat plates". Quart. Appl. Math., 45 (1987), 645-667.

[4] CIARLET, P.G. "A justification of the von Karman equations". Arch. Rat. Mech. Anal., 73 (1980), 349-389.

[5] DUVAUT, G. - LIONS, J.L. "Inequations en thermoelasticite et magneto-hydrodinamique". Arch. Rat. Mech. Anal , 46 (1972), 241-279

[6] HUGHES, T. J. R. - MARSDEN, J E. , "Nonlinear analysis and Mechanics, Heriot-Watt Symposium II", Pitman , 1978, p. 30-285.

[7] RAOULT, A. "Contribution a l'etude des modeles d'evolution de plaques et a l'approximation d'equations d'evolution lineaires de second ordre par des methodes multipas". Thesis, Univ. Pierre et Marie Curie (1980), Paris .

[8] TRABUCHO, L.- VIAÑO, J.M. "Derivation of generalized models for linear elastic beams by asymptotic expansion methods" , in "Applications of Multiple Scaling to Mechanics" (P.G. Ciarlet- E. Sanchez Palencia, eds.), Masson, 1987, Paris.

[9] TRABUCHO, L.- VIAÑO, J.M. "Existence and characterization of higher order terms in an asymptotic expansion method for linearized elastic beams". J. Asympt. Anal.., 2 (1989), 223-255.

A NUMERICAL APPROACH FOR A CLASS OF AXIALLY SYMMETRIC DEFORMATIONS IN NONLINEAR ELASTICITY

E.M. Croitoro and M.F. Pettigrew
Department of Applied Mathematics
The University of Western Ontario
London, Canada, N6A 5B9

Abstract This work is concerned with the deformation, the stress field and the stress concentration in a thick plate with a cylindrical hole or inclusion that is subject to a pressure at the opening while all other boundaries remain traction-free. The equations that model such a deformation are highly nonlinear and coupled. We seek a numerical solution and develop an algorithm based on a Newton-Kantorovich linearization. The equations are discretized with fourth order compact finite differences over a variable grid. The deformation, the stress field and the stress concentration are investigated throughout and are compared with results of linear elasticity theory.

A large class of elastomers, synthetics and some biological tissues obey the theory of Finite Elasticity. The set of partial differential equations that models such materials under stress are usually nonlinear and coupled and few exact analytical solutions are known so far [1],[2].

The problem we are concerned here belongs to a group of boundary value problems involving perturbations about large deformations in thick plates weakened by holes or inclusions. The presence of holes, notches and inclusions results in a stress concentration around the opening and these types of problems are of significant interest in design.

We consider firstly that the plate undergoes a large deformation due to a pressure applied at the opening. The body is further subject to some perturbative surface tractions, reaching a final state of equilibrium. Two types of problems arise: the problem where the thickness of the plate is prevented from changing - this is a two dimensional plane strain problem - and the problem where the thickness is allowed to change and we deal with a three-dimensional case.

We found the general analytical solution [3] for a class of two-dimensional boundary value problems: it applies to incompressible materials with strain-energy function of Mooney-Rivlin type, although the method of solution is not restricted to this particular form of the strain-energy. A typical point located at (ρ,ϑ,ζ) in the unstrained and unstressed state moves to (r,θ,z) after the large deformation takes place. The body is then subject to a certain perturbation which is viewed as a perturbative displacement field of components $\varepsilon u(r,\theta)$ and $\varepsilon v(r,\theta)$ in the r- and θ-direction respectively. The small parameter $\varepsilon \ll 1$ has its own physical significance in each specific problem. We have found all possible combinations of displacement fields $(\varepsilon u,\varepsilon v)$ that can be superimposed on the initial large deformation such that the equilibrium can be maintained without body forces, by surface tractions alone. The solution for $u(r,\theta)$ and $v(r,\theta)$ is expressed in terms of Fourier series where the coefficients themselves are functions of r. The r-dependent functions, in turn, are solutions of a fourth order ordinary differential equation and are expressible in series form with logarithmic contribution.

The general solution is suitable for solving a large class of boundary value problems [3],[4]. A special attention is devoted to the case of a perturbative uniaxial tension applied at great distances from the opening. The section passing through the axis of the hole and which is perpendicular to the line of stretch is the most stressed one. If the body is subject solely to the perturbative uniaxial loading and no previous large deformations occur, the stress at the edge of the hole is three times the stress applied at inifinity - a result obtained within the linear elasticity theory [5]. On the other hand, the body would assume a certain stress field if only a large deformation took place. Now, we consider that the uniaxial tension is applied on the existing finite deformation. We deal here with nonlinearity and we know that superposition doesn't hold. The stress concentration at the hole could be magnified or diminished. We have found a definite increase in stress concentration, as a nonlinear effect. However, the highly stressed region is very localized near the hole and the stress field approaches very rapidly the value prescribed at infinity.

The analysis for the three-dimensional case is more involved. Firstly, we consider that the layer is subject to a pressure applied at the opening, all the other boundaries remain traction-free. The deformation is axially symmetric and is given by the mapping

$$\rho = f(r,\theta), \quad \vartheta = \theta, \quad \zeta = g(r,\theta). \tag{1}$$

While the differential equation that describes the corresponding large deformation in the two-dimensional case can easily be integrated regardless of the functional form of the strain-energy function to obtain $\rho = (r^2 - k^2)^{1/2}$, the set of partial differential equations here are nonlinear and strongly coupled, and to find the analytical solution that satisfies completely and rigorously all the boundary conditions is a difficult task.

Some progress can be made by specifying the strain-energy function. We assume a Neo-Hookean material where $W = C(I_1-3)$.

The equilibrium equations take the form

$$\frac{1}{2C}\frac{\partial p}{\partial r}+\frac{\partial}{\partial r}\left\{\frac{f^2}{r^2}\left[\left(\frac{\partial f}{\partial z}\right)^2+\left(\frac{\partial g}{\partial z}\right)^2\right]\right\}-\frac{\partial}{\partial z}\left[\frac{f^2}{r^2}\left(\frac{\partial f}{\partial r}\frac{\partial f}{\partial z}+\frac{\partial g}{\partial r}\frac{\partial g}{\partial z}\right)\right]$$
$$+\frac{1}{r}\left\{\frac{f^2}{r^2}\left[\left(\frac{\partial f}{\partial z}\right)^2+\left(\frac{\partial g}{\partial z}\right)^2\right]-\left(\frac{\partial f}{\partial r}\frac{\partial g}{\partial z}-\frac{\partial f}{\partial z}\frac{\partial g}{\partial r}\right)^2\right\}=0, \tag{2}$$

$$\frac{1}{2C}\frac{\partial p}{\partial z}-\frac{\partial}{\partial r}\left\{\frac{f^2}{r^2}\left[\left(\frac{\partial f}{\partial z}\right)^2+\left(\frac{\partial g}{\partial z}\right)^2\right]\right\}+\frac{\partial}{\partial z}\left\{\frac{f^2}{r^2}\left[\left(\frac{\partial f}{\partial r}\right)^2+\left(\frac{\partial g}{\partial r}\right)^2\right]\right\}$$
$$-\frac{f^2}{r^3}\left(\frac{\partial f}{\partial r}\frac{\partial f}{\partial z}+\frac{\partial g}{\partial r}\frac{\partial g}{\partial z}\right)=0, \tag{3}$$

where $p(r,\theta)$ is the hydrostatic pressure.

The incompressibility condition requires that

$$\frac{f}{r}\left(\frac{\partial f}{\partial r}\frac{\partial g}{\partial z}-\frac{\partial f}{\partial z}\frac{\partial g}{\partial r}\right)=1. \tag{4}$$

We have found a solution in terms of Bessel functions using a successive approximation method [6]. However, the solution is limited to moderately large deformations.

In order to cast further light on the three-dimensional stress concentration, we seek a numerical solution based on an iterative procedure between f(r,θ), g(r,θ) and p(r,θ).

The formulation for f, g and p involves two coupled second order equations in f and g; it turns out that the nonlinearity appears in the first order terms only. The computational aspects are considerably simplified by introducing the following two transformations:

– the first transformation

$$x = \frac{a}{\rho}, \quad y = \frac{\zeta}{l}, \tag{5}$$

where a is the radius of the cylinder and $2l$ is the plate thickness in the undeformed state, maps the section at infinity into $x = 0$,

– the second transformation given by

$$F(x,y) = \frac{x}{a} f(\rho,\zeta), \quad G(x,y) = \frac{1}{l} g(\rho,\zeta), \tag{6}$$

removes the leading asymptotic behaviour in F.

The domain of computation becomes

$$D : \{ (x,y) \mid 0 \le x \le 1 , \ 0 \le y \le 1 \} . \tag{7}$$

The resulting set of nonlinear equations and the boundary conditions involve the aspect ratio $\gamma = a / l$ which we take as a parameter.

Denote the k th approximate solution by $V^k \equiv [F^k,G^k,P^k]$. We impose a rectangular grid over D with mesh refinement in the vicinity of the opening to accomodate the localized stress concentration. We apply a Newton-Kantorovitch linearization [7],[8] and discretise the linearized equations with variable grid fourth order compact finite differences [9]. The resulting linear algebraic system of equations turns out to be sparse. An iterative sparse matrix solver [10] allows us to obtain F(x,y) and G(x,y). As some terms in the expression for P(x,y) become singular at x=0, we introduce a new function Ψ(x,y) such that

$$\frac{\partial P}{\partial x} = \frac{\partial \Psi}{\partial y}, \quad \frac{\partial P}{\partial y} = - \frac{\partial \Psi}{\partial x}, \tag{8}$$

and hence $\nabla^2\Psi = 0$ in D. From the computation of Ψ(x,y) new values of the hydrostatic pressure and its gradient are obtained. This completes the iteration taking V^k into V^{k+1}. The deformation, the stress field and the stress concentration around the opening is investigated in detail. The computational results are presented graphically and nonlinear effects are highlighted.

Acknowledgements This work has been partially supported by funds provided by the Natural Sciences and Engineering Research Council of Canada.

References

[1] R.W. Ogden, *Nonlinear Elastic Deformations*, Ellis Horwood Series in Mathematics and its Applications, Halsted Press, 1984.

[2] A.E. Green and W. Zerna, *Theoretical Elasticity*, 2nd ed, Oxford University Press, 1968.

[3] E.M. Croitoro, *Perturbctions about Finite Elastic Inflation*, International Journal of Engineering Sciences, Vol. 24, 1986, No. 4, pp.611-629.

[4] E.M. Croitoro and K.A. Lindsay, *Hoop stress calculation for a nearly circular hole*, Journal of Applied Mathematics and Physics, ZAMP, Vol. 35, 1984, No. 6, pp. 865-882.

[5] N.I. Muskhelshvili, *Some Basic Problems of the Mathematical Theory of Elasticity*, 4th ed, Noordhoff, Groningen, 1964.

[6] A.P.S. Selvadurai and A.J.M. Spencer, *Second-order elasticity with axial symmetry: general theory*, International Journal of Engineering Science, 1972, Vol. 10, pp. 97-114.

[7] G. Birkoff and R.E. Lynch, (1984). *Numerical Solution of Elliptic Problems*, SIAM, Philadelphia 1989.

[8] J. Ortego and W. Rheinboldt, *Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[9] M.F. Pettigrew. *On Compact Finite Difference Schemes with Applications to Moving Boundary Problems*, Ph.D. thesis, Applied Mathematics, University of Western Ontario, London, Canada, 1989.

[10] S. Pissanetsky, *Sparse Matrix Technology*, Academic Press, New York, 1984.

ORTHOGONAL WITH NON-INTEGRABLE WEIGHT FUNCTION JACOBI POLYNOMIALS AND THEIR APPLICATION TO SINGULAR INTEGRAL EQUATIONS IN ELASTICITY AND HEAT CONDUCTION PROBLEMS

Igor PODLUBNY
Department of Control Engineering
Faculty of Mining, Technical University
B.Nemcovej 3, 04200 Kosice, Czechoslovakia

Abstract– Two new regularization formulaes for evaluating the finite part of the integrals with non-integrable Jacobi weight function and new spectral relationships for orthogonal with non-integrable weight Jacobi polynomials are obtained. A new approach for solving singular integral equations with Cauchy's kernel, based on the obtained spectral relationships abd regularization formulaes, is proposed. Such approach can be used for the characteristic singular equation as well as for the complete one.

I. INTRODUCTION: THE SHORTEST HISTORY OF THE PROBLEM

1) The definition of the finite part of a divergent integral was given by Hadamard [1], when he was considering the integral

$$I(\lambda) = \int_a^b f(x)(x-a)^{\lambda}dx$$

for $\lambda<-1$. He obtained and used first regularization formulaes for the divergent integtals with the non-integrable weight function.

2) The authors of the Bateman's project [2] noticed that the majority of the relationships for Jacobi polynomials $P_n^{\alpha,\beta}(x)$ can be used even when $\alpha<-1$ or $\beta<-1$ or both $\alpha<-1$ and $\beta<-1$.

3) First application of such Jacobi polynomials was found by Popov and Onishchuk [3]. They reduced a problem for a plate with a rigid inclusion to the integral equation with so called smooth kernel, for which Jacobi polynomials $P_n^{-3/2,-3/2}(x)$, as they proved, are the eigenfunctions. It allowed them to obtain the solution of the integral equation in the form of Fourier-Jacobi series.

II. REGULARIZATION FORMULAES

Lemma 1. If $f(x)\in C^1[-1,1]$, $\alpha<1$, $\beta<1$, then

$$\int_{-1}^{1} \frac{f(t)P_n^{-\alpha-1,-\beta-1}(t)dt}{(1-t)^{\alpha+1}(1+t)^{\beta+1}} = \frac{1}{2n}\int_{-1}^{1}\frac{f'(t)P_{n-1}^{-\alpha,-\beta}(t)dt}{(1-t)^{\alpha}(1+t)^{\beta}}$$

$$(n = \overline{1,\infty})$$

Lemma 2. If $f(x)\in C^1[-1,1]$, $\alpha<1$, $\beta<1$, then

$$\int_{-1}^{1}\frac{f(t)dt}{(1-t)^{\alpha+1}(1+t)^{\beta+1}} =$$

$$= \frac{1}{4\alpha\beta}\int_{-1}^{1}\frac{\alpha-\beta-(\alpha+\beta)t}{(1-t)^{\alpha}(1+t)^{\beta}}f'(t)dt +$$

$$+ \frac{(\alpha+\beta)(\alpha+\beta-1)}{4\alpha\beta}\int_{-1}^{1}\frac{f(t)dt}{(1-t)^{\alpha}(1+t)^{\beta}} .$$

Some particular cases of Lemma 2 are listed below:

$$\int_{-1}^{1}\frac{f(x)dx}{(1-x^2)^{3/2}} = -\int_{-1}^{1}\frac{xf'(x)dx}{(1-x^2)^{1/2}}$$

$$\int_{-1}^{1}\frac{f(x)dx}{(1-x)^{\alpha+1}(1+x)^{1-\alpha}} = -\frac{1}{2\alpha}\int_{-1}^{1}\frac{f'(x)dx}{(1-x)^{\alpha}(1+x)^{-\alpha}}$$

$$\int_{-1}^{1}\frac{f(x)dx}{(1-x)^{\alpha+1}(1+x)^{2-\alpha}} = \frac{1}{4\alpha(1-\alpha)}\int_{-1}^{1}\frac{(2\alpha-1-x)f'(x)dx}{(1-x)^{\alpha}(1+x)^{1-\alpha}}$$

$$(0<\alpha<1)$$

III. SPECTRAL RELATIONSHIPS

Lemma 3. If α is not integer, then

$$\frac{1}{\pi}\int_{0}^{1}\frac{q_n^{m+\alpha,k-\alpha}(\tau)d\tau}{\tau - t} + \frac{q_n^{m+\alpha,k-\alpha}(t)}{tg(\alpha\pi)} =$$

$$= \frac{(-1)^m P_{n+m+k}^{-m-\alpha,-k+\alpha}(1-2t)}{\sin(\alpha\pi)} ;$$

where $q_n^{\alpha,\beta}(\tau) = \tau^{\alpha}(1-\tau)^{\beta}P_n^{\alpha,\beta}(1-2\tau)$,

$n=\overline{0,\infty}$; $k=\overline{0,\infty}$; $m+n+k\geq 0$.

Lemma 4. If α is not integer, then

$$\frac{1}{\pi}\int_0^1 \frac{\tau^{-\alpha+k}(1-\tau)^{\alpha+m}d\tau}{\tau - t} + \frac{t^{-\alpha+k}(1-t)^{\alpha+m}}{tg(\alpha\pi)} =$$

$$= \frac{(-1)^{k+1}P_{m+k}^{\alpha-k,\alpha-m}(1-2t)}{\sin(\alpha\pi)};$$

where $m+k\geq 0$.

IV. THE SOLUTION OF THE CHARACTERISTIC SINGULAR INTEGRAL EQUATION

The use of the previous results is illustrated on the example of the characteristic singular integral equation of the first kind:

$$\frac{1}{\pi}\int_{-1}^{1}\frac{\varphi(t)dt}{t - x} = f(x), \qquad x \in(-1,1) \qquad (1)$$

Using [4], we look for the solution in the following form:

$$\varphi(t)=\sum_{i=1}^{4} A_i w_i(t)+(1-t^2)^{3/2}\sum_{n=0}^{\infty}\varphi_n P_n^{3/2,3/2}(t); \qquad (2)$$

$w_1(t)= w_3(-t)=(1-t)^{-1/2}(1+t)^{3/2}$

$w_2(t)= w_4(-t)=(1-t)^{1/2}(1+t)^{3/2}$.

The constants φ_n $(n=\overline{0,\infty})$ independently on the class of solutions are given by the following expressions:

$$\varphi_n= \frac{f_{n+3}}{8p_{n+3}};\ f_{n+3}= \int_{-1}^{1}\frac{f(x)P_{n+3}^{-3/2,-3/2}(x)}{(1-x^2)^{3/2}}dx$$

$$p_{n+3}= \int_{-1}^{1}\left[P_{n+3}^{-3/2,-3/2}(x)\right]^2(1-x^2)^{-3/2}dx=$$

$$= \frac{\Gamma^2(k + \frac{5}{2})}{4(2k+3)k!(k+3)!}; \qquad (n=\overline{0,\infty}) \qquad (3)$$

The constants A_i $(i=\overline{1,4})$ are dependent on the class of solutions:

a) For the unbounded at the ends of $(-1,1)$ solution under the condition

$$\frac{1}{\pi}\int_{-1}^{1}\varphi(t)dt = 1 \qquad (5)$$

garanting the unique solution we obtain:

$$A_1= -\frac{1}{2} + f_0+ \frac{7}{4} f_1-f_2+ \frac{3f_3}{128p_3}$$

$$A_2= \frac{1}{4} + \frac{1}{2} f_0+ \frac{1}{8} f_1 - \frac{3f_3}{256p} ,$$

$$A_3= 1-f_0- \frac{3}{2} f_1+f_2- \frac{3f_3}{64p_3}$$

$$A_4= \frac{1}{4} + \frac{1}{2} f_0+ \frac{1}{8} f_1- \frac{3f_3}{256p_3}$$

b) For the solution unbounded at the left end and bounded at the right end of $(-1,1)$ we obtain:

$A_1=0,\quad A_2=(f_0+2f_1+f_2)/4,$

$A_3=(f_2-f_0)/2,\qquad A_4= (-3f_0+2f_1+f_2)/4$

c) For the solution bounded at both ends

$A_1=A_3=0,\quad A_2=(f_1+f_0)/2,\quad A_4=(f_1-f_0)/2,$

$$f_k= \frac{1}{\pi}\int_{-1}^{1}\frac{f(x)x^k dx}{(1-x^2)^{3/2}}, \qquad (k=0,1,2)$$

We can find also the solution of the equation (1) in some untraditional classes. For example, for the solution with the bounded at both ends derivative we receive

$A_1=A_2=A_3=A_4=0,$

if the existence conditions are fulfilled [5]:

$f_k=0,\quad k=0,1,2.$

V. CONCLUSIVE REMARKS

The proposed approach can be used also for solving characteristic singular equations of the second kind and for solving complete singular equations of the first and the second kinds with Cauchy's kernel. In the comparison with the famous methods the proposed one has some advantages which are stipulated by the common form of the solution (2)

Finally, if $f(x)\in C^1[-1,1]$, then the series (2) uniformly converges on $[-1+\varepsilon,1+\varepsilon]$, $0<\varepsilon<1$, and $\varphi_n=O(n^{-3/2})$, $n \to \infty$.

REFERENCES:

[1] Hadamard J. Lectures on Cauchy's problem in linear partial differential equations, Yale Univ. Press,1923

[2] Higher transcendential functions,v.2, McGraw-Hill Book Company,New York, 1954

[3] Onishchuk O.V.,Popov G.Ja. Izvestiya Akademii Nauk SSSR, Mekhanika tverdogo tela, 4(1980),141-150 (in Russian)

[4] Podlubny I. On the behaviour of the Cauchy's type integral when the density function vanishes at the ends of the contour of integration, Odessa, 1989, deposited in the All-Union Institute of Scientific and Technical Information 22.12.1989, reg.number 7603-V89,(in Russian).

[5] Farshajt P.G. The solution of the boundary value problems for the biharmonic equation with linear defects presenting in a domain, doctoral thesis, Odessa,Odessa State Univ., 1989 (in Russian)

FINITE DIFFERENCE SIMULATION OF TIME-DEPENDENT FLOW AROUND A LONGITUDINALLY OSCILLATING CYLINDER

PAPOLU MANIKYALA RAO
Prof.M. Kawahara Laboratory,Department of Civil Engg,
CHUO University,Kasuga,Bunkyo-ku,Tokyo,Japan.

AND

KUNIO KUWAHARA
The Institute of Space and Astronautical Science
Sagamihara-Shi,Kanagawa,229,Japan

ABSTRACT- A finite difference solution is obtained for the time dependent viscous incompressible 2-dimensional flow past a longitudinally oscillating circular cylinder.A Navier-Stokes equation of finite difference form is solved by moving grid system,based on a time-dependent coordinate transformation.The solution describes the development of the vortex street developed behind the cylinder when cylinder remains stationary.The time-dependent lift and drag coefficients are obtained.The power spectra of the drag,lift and displacement is also reported.The computer results predict the lock-in phenomenon which occurs when oscillation frequency is around double the natural vortex sh dding frequency.

INTRODUCTION

The understanding of the characteristics of vibrations induced by vortex shedding is of great importance in the design of structures such as heat exchangers,offshore platforms,power cables,etc.The determination of the forces acting on a cylinder undergoing harmonic in-line oscillations is of special interest in aeroelasticity as well as for a basic understanding of the fluid mechanics. Information about the in-line oscillations of a cylinder in uniform flow has not been studied extensively.However,only a few studies have been made for a circular cylinder oscillating in a uniform flow[1-4].

The numerical simulation of steady flow past a circular cylinder undergoing in-line and/or transverse oscillations through the use of two-dimensional unsteady Navier-Stokes equations was taken by [5] for relatively small amplitudes.Recently,numerical solutions of unsteady flow about stationary and oscillati cylinder has been obtained[6-9].

THE GOVERNING EQUATIONS AND NUMERICAL METHOD

The non dimensional governing Navier Stokes equations, written in non conservative form and expressed in the dimensionalless quantities,are

$$\mathrm{div}\mathrm{V} = 0 \qquad (1)$$

$$\frac{\partial \mathrm{V}}{\partial t} + (\mathrm{V}\cdot\nabla)\mathrm{V} = -\mathrm{grad}p + \frac{1}{Re}\Delta \mathrm{V} \qquad (2)$$

where $\mathrm{V} = (u,v)$ is the velocity,and p the pressure. Re denotes the Reynolds number based on the velocity of the uniform flow, U, and the diameter, d, of the circular cylinder.The reference scales for non-dimensionalization were $d, U, d/U, \rho U^2$ for the length,velocity,time and pressure,respectively.

The numerical techniques adopted here are based on the well-known Marker and Cell(MAC)method,which was developed by [10].The Poisson equation for the pressure,derived by taking the divergence of (2). All the spatial derivatives except those of the convection terms are approximated by second-order central differencing scheme.The non-linear terms are represented by means of a third order upwind scheme[11],

$$(U\frac{\partial f}{\partial x})_i = \begin{array}{ll} U_i(f_{i+2} - 2f_{i+1} + 9f_i - 10f_{i-1} + 2f_{i-2})/6h & (U_i > 0) \\ U_i(-2f_{i+2} + 10f_{i+1} - 9f_i + 2f_{i-1} - f_{i-2})/6h & (U_i \leq 0) \end{array} \qquad (3)$$

where h is grid scale.The Poisson equation for the pressure is solved iteratively by employing a modifed SOR method. For the marching Navier Stokes equation, the semi implicit scheme which is equivalent to the Euler backward scheme is used to integrate temporally.

In this paper,the following co-ordinate transformations, which includes the time variable are introduced

$$\xi = \xi(x,y,t)\ ,\ \ \eta = \eta(x,y,t)\ ,\ \ \tau = \tau(x,y,t)\ , \qquad (4)$$

here x and y are the Cartesian co-ordinates and t is the time in the physical domain, ξ and η are the space co-ordinates and τ the time in the computational domain.

The transformation of the coordinate derivatives are given by

$$\begin{bmatrix} \partial_x \\ \partial_y \\ \partial_t \end{bmatrix} = \begin{bmatrix} \xi_x & \eta_x & \tau_x \\ \xi_y & \eta_y & \tau_y \\ \xi_t & \eta_t & \tau_t \end{bmatrix} \begin{bmatrix} \partial_\xi \\ \partial_\eta \\ \partial_\tau \end{bmatrix} \qquad (5)$$

where subscripts denote the partial derivatives.The time variable is treated as

$$\tau = t\ ,\ \ t_\xi = t_\eta = 0 \qquad (6)$$

Substituting equation(6) into equation(5),one can obtain

$$J\begin{bmatrix} \partial_x \\ \partial_y \\ \partial_t \end{bmatrix} = \begin{bmatrix} y_\eta & -y_\xi & 0 \\ -x_\eta & x_\xi & 0 \\ x_\eta y_\tau - y_\xi x_\tau & y_\xi x_\tau - x_\xi y_\tau & J \end{bmatrix} \begin{bmatrix} \partial_\xi \\ \partial_\eta \\ \partial_\tau \end{bmatrix} \qquad (7)$$

where (x_τ, y_τ) denotes the velocity vector of a grid point and J is the Jacobian for the space coordinate transformation. The convective terms in the Navier Stokes equations will be expressed as;

$$\partial_t + u\partial_x + v\partial_y = \partial_\tau + \frac{1}{J}[\{(u-x_\tau)y_\eta - (v-y_\tau)x_\eta\}\partial_\xi$$

$$+\{(v-y_\tau)x_\xi - (u-x_\tau)y_\xi\}\partial_\eta] \qquad (8)$$

the transformations given by the equation(8) implies that the moving boundaries in the physical domain become stationary in the computational domain.

RESULTS AND DISCUSSION

The entire computation were run on a Fujitsu FACOM VP200 Super Computer for the Reynolds number,Re= 1000. The results of the stationary circular cylinder are presented first to allow comparison with the experimental visualization.The results of the unsteady flow for a cylinder oscillates in the longitudinal direction are then presented. Figure 1(a) represents the development of the flow pattern computed for a stationary cylinder at time t=100.The development of the vortex shedding is clearly seen.Two secondary vortices are formed,followed by a smaller single down stream of the separation point.As the vortices grow symmetrically,the shear layer joining the separation point to one of the vortices begins to develop instabilities and is drawn across the wake in response to the base pressure reduced by the action of the vortex growing across the wake.The stretching,diffusion,and dissipation of vorticity break up the deforming turbulent sheet and thereby the further supply of circulation to the vortex whose rate of growth has already been reduced to its minimum.The vortex across the wake still continues to grow and entrains part of the oppositely signed vorticity.The shedding process for the second vortex does not commence until the circulation in its feeding sheet decreases to its minimum,making the sheet most susceptible to rapid diffusion.The eddies that develop in the near wake during a given period undergo successive interactions among themselves.This configuration agrees with the experimental and numerical results.The unsteady lift coefficient oscillates periodically.The perodic properties of the flow are clearly shown by the time dependent evolution of the lift coefficient as shown in figure 1(b).

The numerical solutions of longitudinal oscillations of circular cylinder was performed.These results are for the case of the reduced velocity V_r =3.0,5.0,5.25,7.2.Here, V_r is defined as $V_r = U/f_c d, U = U_m$ where U_m is the maximum velocity of the oscillating cylinder,f_c the cylinder frequency. The amplitude of oscillation was set to be $A(x/d)$ = 0.14,0.17,0.2,0.26.,where x refers to the displacement in x-direction.Figure 1(c) explains the flow visulization of the wake of oscillating cylinder reveals different patterns than the usual alternate Karman vortex street.In the lock-in range the influence of the flow oscillation is very strong.During one period of the flow oscillation,the vortex remains close to the cylinder as it forms and grows,even close in fact than before locking-on.The interesting feature of the phenomenon is the cylinder moves down stream from its mid position,the relative velocity about the cylinder increases from zero to U.As the cylinder reverses its direction,the relative velocity increases and reaches a value of 2U at the mid position of oscillation.The vortices move away from the cylinder under the influence of the uniform flow,symmetrically at first and then becoming gradually asymmetrical. The resulting pattern is very complex .The streamwise interval of the vortices becomes large.Accordingly the value of lift is changing temporally and becomes complicated with including disturbances and different from the simple sinusoidal curve and are shown in figure 1(b)and 2.The mean inline force and its amplitude of oscillation were found to increase with increasing amplitude.The power spectra of the drag,lift and displacement is illustrated in figure 3. Synchronization can be seen when the driving frequency approaches in a range around double the Strouhal frequency.

REFERENCES

1.N.Ferguson and G.V.Parkinson, ASME J.Engg.Industry 89, 831(1967).

2.Y.Tanida, A.Okajima and Y.Watanabe, J.Fluid Mech 61, 769(1973).

3.O.M.Griffin and S.E.Ramberg, J.Fluid Mech 75, 257(1976).

4.C.Barbi, D.F.Favier., et.al, J.Fluid Mech 170, 527(1986).

5.Y.Lecointe, J.Piquet and J.Plantec, Flow structure in the Wake of an oscillating cylinder,"Forum on unsteady flow separation(Ed.K.N.Ghia),ASME FED-Vol.52,147(1987)

6.P.M.Rao, K.Kuwahara and K.Tsuboi,-Submitted to Applied Math. Modelling.

7.P.M.Rao, K.Kuwahara and K.Tsuboi,A direct simulation of the flow around a circular cylinder sinusoidally oscillating at low Keulegan-Carpenter number- Accepted 13th IMACS,Dublin.

8.P.M.Rao, K.Kuwahara and K.Tsuboi,-Submitted to Journal of Fluid Mechanics

9.P.M.Rao, K.Kuwahara,Procd. of Indian Academy of Sciences(1991) (in press)

10.F.H.Harlow and J.E.Welch, Phys.Fluids 8,2182(1965).

11.T.Kawamura and K.Kuwahara,AIAA paper 84-0340(1984).

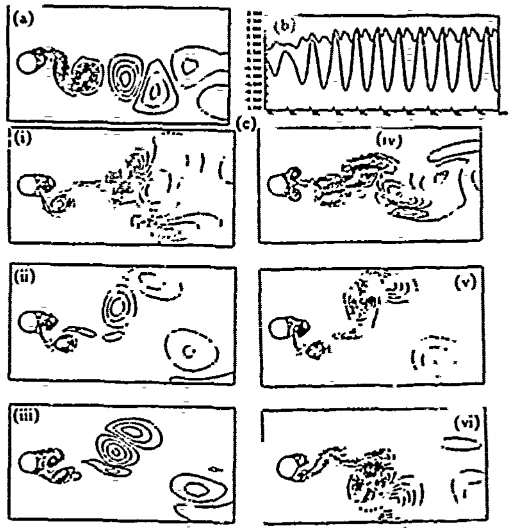


Figure.1. For a stationary cylinder. (a)Vorticity contour, (b)Time dependent lift and drag; (c)Vorticity contour for an oscillating cylinder;
(i) A=0.14, V_r=5.0,(ii) A=0.17, V_r=5.0,(iii) A=0.26, V_r=5.25,(iv) A=0.2, V_r=3.0, (v) A=0.2, V_r=5.0,(vi) A=0.2 V_r=7.2.

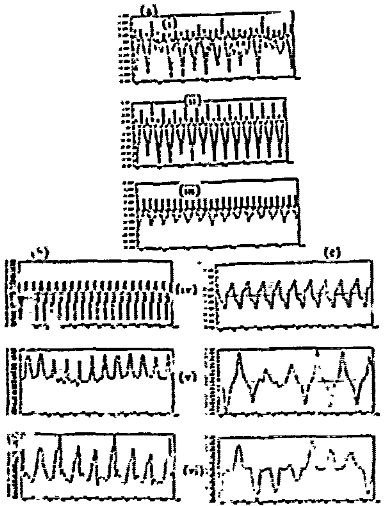


Figure 2 (a) Time dependent lift and drag for an oscillating cylinder. (i) A=0.14, V_r=5.0,(ii) A=0.17, V_r=5.0, (iii)A=0.26, V_r=5.25 (b) Drag ,(c) Lift A=0.2, (iv) V_r=3.0, (v) V_r=5.0 (vi) V_r=7.2

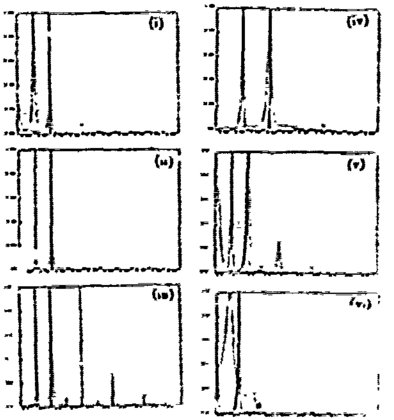


Figure.3 Power spectra of the lift and drag for an oscillating cylinder. (i) A=0.14, V_r=5.0,(ii) A=0.17, V_r=5.0, (iii)A=0.26, V_r=5.25, A=0.2 (iv) V_r=3.0 (v) V_r=5.0,(vi) V_r=7.2

CONVECTIVE HEAT TRANSFER DUE TO A ROTATING DISK SUBJECTED TO A MIXED THERMAL BOUNDARY CONDITION

G. RAMANAIAH
Dean of Science and Humanities
Anna University
Madras-600 025 INDIA

and

V. KUMARAN
Department of Mathematics
Anna University
Madras-600 025 INDIA

Abstract : The paper deals with flow and heat transfer induced by a rotating circular disk when it is subjected to a mixed thermal boundary condition. By similarity analysis the governing equations are reduced to a system of ordinary differential equations. A transformation group has been found under which the reduced governing equations and the homogeneous boundary conditions are invariant. The thermal boundary condition is characterized by a nonnegative parameter m; m=0,1,∞ correspond to the cases of prescribed temperature, prescribed heat flux and prescribed heat transfer coefficient (radiation boundary condition) respectively. If one wants to find solutions for different values of m, there is no need to compute solution for each value of m. If we know the solution for any particular value of m, then this solution can be used to find the solution for any value of m by a simple method with the aid of the transformation group.

I. INTRODUCTION

The free convection and mixed convection on heated horizontal plates has received a great deal of attention [1-6]. These studies assume that the temperature at the plate is prescribed or the heat flux across the plate is prescribed. The study of convection problem when the plate is subjected to a mixed thermal boundary condition [7] has not received sufficient attention. The axisymmetric forced convection due to a rotating disk was studied first by Von Karman [8] and later Cochran [9] improved the numerical solution. In this paper we present an analysis of the convection problem induced by a rotating disk which is subjected to a mixed thermal boundary condition.

II. ANALYSIS

The convection over a rotating horizontal disk subjected to a mixed thermal boundary condition is governed by the boundary layer equations,

$$\frac{\partial}{\partial r}(ru) + \frac{\partial}{\partial z}(rw) = 0 \qquad (1)$$

$$u\frac{\partial u}{\partial r} + w\frac{\partial u}{\partial z} - \frac{v^2}{r} = -\frac{1}{\rho_\infty}\frac{\partial p}{\partial r} + \gamma\frac{\partial^2 u}{\partial z^2} \qquad (2)$$

$$u\frac{\partial v}{\partial r} + w\frac{\partial v}{\partial z} + \frac{uv}{r} = \gamma\frac{\partial^2 v}{\partial z^2} \qquad (3)$$

$$\frac{1}{\rho_\infty}\frac{\partial p}{\partial z} = g\beta(T-T_\infty) \qquad (4)$$

$$u\frac{\partial T}{\partial r} + w\frac{\partial T}{\partial z} = \frac{\gamma}{Pr}\frac{\partial^2 T}{\partial Z^2} + \frac{\gamma}{c}\left[\left(\frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial z}\right)^2\right] \qquad ...(5)$$

with the boundary conditions,

$u = 0, v = r\omega, \quad w = 0 \quad$ at $z = 0$ (6)

$u \to 0, v \to 0, p \to p_\infty, T \to T_\infty$ as $z \to \infty$ (7)

$$a_0(T-T_\infty) - a_1\frac{\partial T}{\partial z} = a_2 r^2 \text{ at } z = 0 \qquad (8)$$

where u,v and w are the velocity components in the radial, azimuthal and axial directions respectively, T is the temperature, p is the pressure, p_∞, T_∞, ρ_∞ are the ambient pressure, temperature and density respectively. ω is the angular velocity of the rotating disk. The symbols γ,g,ß,c and Pr denote the kinematic viscosity, gravitational acceleration, coefficient of thermal expansion, specific heat and Prandtl number respectively, a_0, $a_1 \geq 0$, $a_2 \geq 0$ are prescribed constants.

Let us introduce the similarity transformation

$$\eta = z/L,\ u = \frac{\gamma r}{L^2}F'(\eta),\ v = \frac{\gamma r}{L^2}G(\eta),\ w = -\frac{2\gamma}{L}F(\eta)$$
$$p - p_\infty = -\frac{\rho_\infty \gamma^2 r^2}{L^4}H(\eta),\ T - T_\infty = \frac{r^2}{\lambda L^5}\theta(\eta) \qquad (9)$$

where $\lambda = g\beta/\gamma^2$ and L is of dimension length to be determined from the thermal boundary condition (8) in the maner as explained a little later. The equations (1) to (8) become

$F''' + 2FF'' - F'^2 + G^2 + 2H = 0$ (10)
$G'' + 2(FG'-F'G) = 0$ (11)
$H' + \theta = 0$ (12)
$\theta'' + 2Pr(F\theta' - F'\theta) + Pr\,E\,(G'^2+F''^2) = 0$ (13)

$F(0) = F'(0) = 0, G(0) = \Omega$ (14)
$F'(\infty) = G(\infty) = H(\infty) = \theta(\infty) = 0$ (15)
$(1-m)\,\theta(0) - m\theta'(0) = 1$ (16)

where the primes denote differentiation with respect to η.
$\Omega = \omega L^2/\gamma$ is the rotation parameter (17)
$E = \lambda L\gamma^2/c$ is the Eckert number (18)
L is the positive root of the equation,
$a_2\lambda L^6 - a_0 L - a_1 = 0$ (19)
and $m = a_1/(a_1+La_0)$.

The equation (19) has a unique positive root by Descartes rule of sign for $a_1 \geq 0$, $a_2 \geq 0$. The solution of the boundary value problem (10)-(16) can be obtained by shooting method, for any value of $m \geq 0$. It is interesting to note that the solutions corresponding to different values of m are dependent as stated in the following properties :

Property 1 : The equations (10)-(15) are invarient under the transformation,
$\eta^* = A\eta$, $\Omega^* = \Omega/A^2$, $E^* = E/A$
$F^*(\eta^*,\Omega^*,E^*) = F(\eta,\Omega,E)/A$, $G^*(\eta^*,\Omega^*,E^*) = G(\eta,\Omega,E)/A^2$ (20)
$H^*(\eta^*,\Omega^*,E^*) = H(\eta,\Omega,E)A^4$, $\theta^*(\eta^*,\Omega^*,E^*) = \theta(\eta,\Omega,E)/A^5$
where A is any positive real number.

Property 2 : If $F(\eta,\Omega,E)$. $G(\eta,\Omega,E)$. $H(\eta,\Omega,E)$ and $\theta(\eta,\Omega,E)$ is the solution of the boundary value problem (10)-(16) for any particular value of m, say m_0, then the solution for any value of m is given by eqns. (20) provided A is the positive root of the equation,
$A^6 - (1-m)\,A\,\theta(0,\Omega,E) + m\,\theta'(0,\Omega,E) = 0$ (21)

Property 3 : If the solution of the boundary value problem (10)-(16) is same for any two distinct values of m, then the solution is same for all values of m.

III. DISCUSSION AND CONCLUSION

The study of convection over a horizontal rotating disk subjected to a mixed thermal boundary condition has been reduced to solving the boundary value problem (10)-(16). The boundary condition (16) includes the following as special cases :
(1) Prescribed temperature (PT)
Here $a_0 > 0$, $a_1 = 0$, $a_2 > 0$
Hence $L = (a_0/\lambda a_2)^{1/5}$, m=0 and equation (16) becomes
$\theta(0) = 1$ (16a)
(2) Prescribed heat flux (PHF)
Here $a_0 = 0$, $a_1 > 0$, $a_2 > 0$
Hence $L = (a_1/\lambda a_2)^{1/6}$, m=1 and equation (16) becomes
$\theta'(0) = -1$ (16b)

(3) Prescribed heat transfer coefficient (PHTC)
Here $a_o < 0$, $a_1 > 0$, $a_2 = 0$.
Hence $L = -a_1/a_o$, $m = \infty$ and equation (16) becomes

$$\theta(0) + \theta'(0) = 0 \tag{16c}$$

The local Nusselt number Nu$_r$, defined by

$$Nu_r = -\frac{r}{(T-T_\infty)}\frac{\partial T}{\partial z}\Big|_{z=0} \text{ becomes}$$

$$Nu_r = -\frac{r}{L}\frac{\theta'(0)}{\theta(0)}$$

$$= (\lambda a_2/a_0)^{1/5}\, r\, [-\theta'(0)] \qquad \text{for } PT$$

$$= (\lambda a_2/a_1)^{1/6}\, r/\theta(0) \qquad \text{for } PHF$$

$$= |\; a_o \;|\; r/a_1 \qquad \text{for } PHTC$$

The numerical results for different values of the parameters Ω,E,m and Pr=0.72, are given in Tables I and II. All the quantities of interest increase with m. The temperature profiles are shown in Figures 1(a)-1(c). It is found that in the absence of dissipation, increase in rotation (Ω) decreases the temperature, whereas increase in dissipation (E) increases the temperature near the disk and it is more pronounced with Ω. Increase in m increases the temperature largely, and decreases the thermal boundary layer thickness. Table III gives the values of A needed for transition from one case to the other with the aid of equation (20). The following example illustrates the transition.

For PT with Ω=1, E = 0.1
$\theta'(0) = -0.7451$, $F''(0) = 1.3722$
$G'(0) = -0.9284$, $H(0) = 0.9738$
which gives

$A = 0.9521$, $\Omega = 1.1031$, $E = 0.1050$
$\theta(0) = 1.2779$, $F''(0) = 1.5897$ for PHF
$G'(0) = -1.0755$, $H(0) = 1.1848$

$A = 0.7451$, $\Omega = 1.8012$, $E = 0.1342$
$\theta(0) = 4.3544$, $F''(0) = 3.3172$ for PHTC
$G'(0) = -2.2444$, $H(0) = 3.1595$

It is interesting to know that the solution is independent of m for certain values of Pr, Ω and E. Table IV shows the values of Pr for E=0, for which the solution is independent of m.

ACKNOWLEDGEMENT : One of the authors (V.Kumaran) is thankful to Council of Scientific and Industrial Research, New Delhi, India for the financial assistance.

Table I
Values of $\theta(0)$, $-\theta'(0)$, $-\theta'(0)/\theta(0)$ for various values of Ω,E and m

Ω	E	m	$\theta(0)$	$-\theta'(0)$	$-\theta'(0)/\theta(0)$
0.0	0.0	0	1.0000	0.7570	0.7570
		0.5	1.1266	0.8734	0.7753
		1	1.2610	1.0000	0.7930
		∞	4.0214	4.0214	1.0000
	0.1	0	1.0000	0.7410	0.7410
		0.5	1.1360	0.8640	0.7605
		1	1.2827	1.0000	0.7796
		∞	4.3543	4.3543	1.0000
0.1	0.0	0	1.0000	0.7575	0.7575
		0.5	1.1263	0.8737	0.7757
		1	1.2606	1.0000	0.7933
		∞	4.0179	4.0179	1.0000
	0.1	0	1.0000	0.7411	0.7411
		0.5	1.1360	0.8640	0.7606
		1	1.2825	1.0000	0.7797
		∞	4.3527	4.3527	1.0000

Table II
Values of $F''(0)$, $-G'(0)$ and $H(0)$ for various values of Ω,E and m

Ω	E	m	$F''(0)$	$-G'(0)$	$H(0)$
0.0	0.0	0	1.0966	0	0.9933
		0.5	1.1778	0	1.0927
		1	1.2603	0	1.1958
		∞	2.5273	0	3.0241
	0.1	0	1.1023	0	0.9976
		0.5	1.1898	0	1.1046
		1	1.2795	0	1.2171
		∞	2.6611	0	3.2329
0.1	0.0	0	1.0992	0.0878	0.9929
		0.5	1.1803	0.0899	1.0921
		1	1.2625	0.0919	1.1951
		∞	2.5280	0.1159	3.0216
	0.1	0	1.1051	0.0880	0.9973
		0.5	1.1924	0.0903	1.1043
		1	1.2821	0.0925	1.2168
		∞	2.6626	0.1180	3.2317

Table III
Values of A for Transition

→	PT	PHF	PHTC
PT	1	$[-\theta'(0,\Omega,E)]^{1/6}$	$-\theta'(0,\Omega,E)$
PHF	$[\theta(0,\Omega,E)]^{1/5}$	1	$1/\theta(0,\Omega,E)$
PHTC	$[\theta(0,\Omega,E)]^{1/5}$	$[\theta(0,\Omega,E)]^{1/6}$	1

Table IV
Critical Values of Pr for E = 0

Ω	$F''(0)$	$-G'(0)$	$H(0)$	Pr
0.0	0.6570	0.0000	0.7100	2.4372
0.1	0.6616	0.0669	0.7101	2.4215
0.5	0.7653	0.3552	0.7138	2.1213
1.0	1.0481	0.8004	0.7252	1.6273

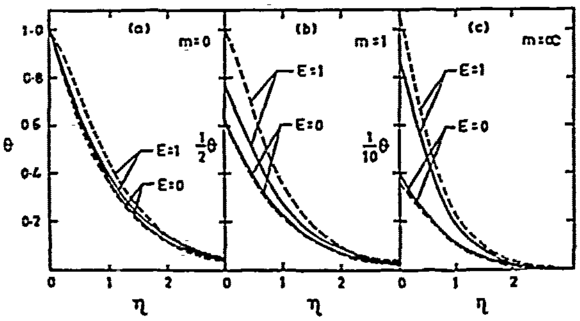


Fig.1 The temperature profiles for (a) PT (b) PHF (c) PHTC
(—— Ω = 0, ------ Ω=1)

REFERENCES

1. Stewartson, K., ZAMP. 9, 276-281 (1958).
2. Gill, W.N., Zeh, D.W., Casel E.D., ZAMP, 16, 539-541 (1965).
3. Clarke,J.F., Riley,N., Q. Jl. Mech. Appl. Math., 28, 373-396 (1978).
4. Clarke, J.F., Riley, N., J. Fluid Mech., 74, 415 431 (1976).
5. Schneider, W., Int. J. Heat Mass Transfer, 22, 1401-1406 (1979).
6. Merkin, J.H., Ingham, D.B., ZAMP, 38, 102-116 (1987).
7. Malarvizhi, G., Ramanaiah, G., Vol.2, 77-86, proceeding of the first international conference on Advanced computational methods in heat transfer, Computational Mechanics Publications, Southampton, Boston (1990).
8. Karman, Th. V., Z. angew. Math. Mech., 1, 233-51 (1921).
9. Cochran, W.G., Proc. Camb. Phil. Soc., 30, 365-75 (1934).

TIME AND SPACE SELF-ADAPTIVE NUMERICAL METHODS FOR CHEMICALLY REACTING FLOWS

B. Rogg

University of Cambridge, Department of Engineering,
Trumpington Street. Cambridge CB2 1PZ, UK

Abstract Self adaptive numerical solution methods for combustion problems are developed and results are presented for physically idealized situations. First, the treatment of the time-dependence is discussed separately, as is the development and implementation of spatial adaptive-gridding techniques, representative numerical results based these two major ingredients of any numerical technique for the simulation of ignition problems are presented. Secondly, time-dependence and spatially two-dimensional variations are treated simultaneously; as a representative example self-ignition in a premixed reactive mixture is numerically simulated.

1. INTRODUCTION

In combustion and related areas numerical techniques are needed which are able to produce high spatial and temporal resolution of flame structures and evolutions; only such ability will provide enhanced insight into the many modes and ingredients of combustion processes. The use of adaptive methods, i.e., of methods that "automatically" adapt the computational mesh in some "optimal" way to the specific problem under consideration, is the desirable numerical approach to all engineering problems; it is a must, however, for the computation of reactive flows. The physical reason for this is the heat-release associated with any combustion process. As a consequence of the heat release, profiles of quantities involved in combustion processes typically exhibit steep gradients and strong curvature, thus necessitating the use of adaptive methods in order to control both the temporal and spatial discretization errors. For time-independent steady problems in combustion, adaptive methods have proven useful and are being used, see e.g. /1 3/. Apart from notable exceptions /4.5/, little attention has been given to time dependent, two-dimensional combustion problems. Therefore, in the present paper we outline possible approaches to fully implicit adaptive algorithms suitable for successfully tackling the numerical simulation of unsteady, spatially two-dimensional reactive flows.

Firstly, we consider briefly techniques suitable for tackling the time-dependence of combustion phenomena and present two spatially one-dimensional examples, viz., a pulsating flame and a self ignition process taking place in a non premixed flow. Secondly, we consider spatially two dimensional, steady problems focusing attention on the development of self adaptive meshing procedures. As a representive example, numerical results are presented for a laminar flame propagating in a strained mixing layer. Thirdly, we combine the seperately developed ingredients for the simulation of time-dependent and spatial effects, and consider truly time-dependent combustion in two space dimensions. For the latter case, hot-spot-like self-ignition taking place in an originally cold, reactive mixture provides the physical basis for a numerical example.

2. FORMULATION

2.1 Differential equations

The general class of differential equations to be considered herein can be written in the form

$$
\begin{aligned}
&\mathbf{u}_t = \mathbf{f}(x,y,t,\mathbf{u},\mathbf{u}_x,\mathbf{u}_y,\mathbf{u}_{xx}\mathbf{u}_{xx},\mathbf{u}_{xy}) \text{ for } (x,y) \in \Omega, \quad t>0,\\
&\mathbf{r}(x,y,t,\mathbf{u}(x,y,t),\mathbf{u}_x(x,y,t),\mathbf{u}_y(x,y,t)) = 0\\
&\qquad\qquad\qquad\qquad \text{for } (x,y) \in \Omega \cup \partial\Omega, \quad t>0,\\
&\mathbf{u}(x,y,0) = \mathbf{v}(x,y) \text{ for } (x,y) \in \Omega \cup \partial\Omega,
\end{aligned}
\tag{1}
$$

where u, f, r and v are N-vectors, x and y are the cartesian space coordinates and t the time; Ω denotes a rectangular domain of integration and $\partial\Omega$ its boundary. Equation (1) represents a nonlinear parabolic mixed initial boundary value problem for the N dependent variables represented by u.

2.2 Difference equations

The integration of system (1) with respect to time is performed in steps starting with specified profiles [which, in general, should satisfy the governing equations] at time level $n = 0$ with $t = t^0 \doteq 0$. Solutions to (1) are sought at the subsequent time levels ($n = 1, t = t^1$), ($n = 2, t = t^2$), and so on, with $0 = t^0 < t^1 < t^2 < \cdots < t^n < \cdots$, where here and below the superscript n is used to identify quantities at time level n, $n = 0,1,2,$. The integration of (1) is considered complete if either a specified level n_{max} or a specified time t_{max} is reached. With respect to the space variables x and y, system (1) is discretized on a mesh M^n of grid points,

$$
M^n = \{(x_1^n, y_1^n), (x_2^n, y_2^n)^n \cdots (x_m^n, y_m^n)\}. \tag{2}
$$

Note that the mesh may or may not be a simple tensor-product grid. In fact, for the simulation of combustion problems irregular grids are desirable which locally and instantaneously concentrate grid points where they are needed, namely in regions or spots of high spatial and temporal activity. A suitable adaptive-meshing strategy is outlined below. If the spatial domain of integration is infinitely large with respect to one or both coordinate directions, for example extending from $x = -\infty$ to $x = +\infty$, then $-a_s$ and a_e are chosen as positive and sufficiently large to ensure that the conditions at either boundary can asymptotically be satisfied /6/, an infinite domain of integration with respect to y is treated similarly.

3. NUMERICAL METHODS

Subsequently the symbols U and F will be used to denote the $m \times N$-vectors which result from the spatial discretization of the N-vectors u and f, respectively. For the solution of time-dependent problems at each time step, or steady-state problems (not considered herein), Newton's method can be applied to the system of nonlinear equations, F(U), which results from the discretization of the governing equations. Note that both U and F depend on the particular time level n under consideration. Thus, the linear system

$$J(\mathrm{U}^k)(\mathrm{U}^{k+1} - \mathrm{U}^k) = -\omega_k \mathrm{F}(\mathrm{U}^k), \quad k = 0, 1, \dots \; , \qquad (4)$$

is solved where U^k denotes the solution after k Newton iterations, and ω_k and $J(\mathrm{U}^k)$ are the damping parameter and the Jacobian matrix, respectively, based on U^k. The damping strategy /7,8/ allows the Jacobian, which is generated numerically, to be re-evaluated only periodically /8/.

Upon discretising the governing equations only with respect to space, one is left with a large systems of DAEs (differential-algebraic equations). In recent years various algorithms have been developed for the solution of such systems as well as complete software packages. Examples for the latter are *DASSL* /9/ and *LIMEX* /10,11/ developed at the Sandia National Labs., USA, and at the University of Heidelberg, West Germany, respectively. *DASSL* uses backward-differentiation formulas, *LIMEX* a newly developed extrapolation method particularly designed to deal with stiff systems. Since, in a sense, software packages are multi-purpose codes, they are not optimized with regard to a specific problem under consideration. Therefore, we have developed a new code particularly designed for the solution of systems of DAEs that arise in one-dimensional combustion problems which, however, uses some of the basic elements of *LIMEX*.

Regardless whether Newton's method or the method of lines is employed, the solution of system (1) at time level n depends on the solution at level $n-1$ taken at the grid points of mesh M^n. Since in general the grids at levels $n - 1$ and n are not the same, the solution obtained at level $n-1$ on grid M^{n-1} must be interpolated onto grid M^n which, regardless of the interpolation procedure, introduces an additional spatial discretization error into the algorithm. We use piecewise monotonic cubic Hermite interpolation /12/.

4. ADAPTIVE SELECTION OF GRID POINTS

The procedures and criteria for the adaptive selection of grid points are of critical importance to the efficiency of the algorithms that are used for the solution of combustion problems. In particular, strategies are required that place the grid points where they are needed in order to bound the local space discretization error. In generalization of procedures outlined previously for the adaptive computation of steady one dimensional combustion problems, for any fixed time level n we equidistribute the mesh M^n on intervals h_l^n, k_l^n with respect to a non negative weight function W^n and a constant C^n. Here the intervals in the x and y direction, h_l^n and k_l^n, are of lengths characteristic of the local and instantaneous spacing of mesh points in the vicinity of point (x_l^n, y_l^n). For instance, for the x direction W^n is selected such that

$$\int_{h_l^n} W^n dx = C^n, \qquad (5)$$

where $m = m^n$. Specifically, the weight function W^n is chosen as

$$W^n = \max_{1 \le k \le 2N+1} W_k^n, \qquad (6.a)$$

where

$$W_k^n = \frac{|\partial U_k / \partial x|}{g^n |\max U_k - \min U_k|}, \quad 1 \le k \le N \qquad (6.b)$$

$$W_{N+k}^n = \frac{|\partial^2 U_k / \partial x^2|}{c^n |\max(\partial U_k / \partial x) - \min(\partial U_k / \partial x)|}, \quad 1 \le k \le N, \qquad (6.c)$$

and

$$W_{2N+1}^n = d^n. \qquad (6.d)$$

In Eqs. (6.b) and (6.c) "min" and "max" stand for the minimum and maximum value in the interval $a_s \le x \le a_e$ of the respective quantity, and g^n and c^n are positive scaling factors; their numerical values are less than unity if in Eq. (5) $C^n = 1$ is employed. In Eq. (6.d), d^n is a positive constant which represents the maximum size of any interval h_l. To prevent the size of adjacent mesh intervals from varying too rapidly, we require that at any time level n the mesh be locally bounded ,viz.,

$$R^{-1} \le h_j / h_{j-1} \le R, \qquad (7)$$

where R is a constant greater than one. The equidistribution procedure with respect to the y direction is performed analogously. The adaptive gridding procedure to be carried out at *each* time level n essentially consists of 7 steps which, because space is restricted in the present volume, can be presented only in the full-length version of the paper.

5 EXAMPLES, RESULTS AND DISCUSSION

As a first example we consider a problem, the so-called test problem A, proposed for a GAMM Workshop at Technical University Aachen, West Germany, viz., an unsteadily propagating flame with one-step chemistry and Lewis number different from unity /13/. The governing equations are

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + R, \qquad (8)$$

$$\frac{\partial Y}{\partial t} = \frac{1}{Le} \frac{\partial^2 T}{\partial x^2} - R, \qquad (9)$$

where T and Y denote a normalized temperature and mass fraction, respectively, and where

$$R = \frac{\beta^2}{2Le} Y \exp\left(-\frac{\beta(1-T)}{1-\alpha(1-T)}\right). \tag{10}$$

Time t and space variable x are suitably nondimensionalized. In the rate expression (10), α and β denote a nondimensional heat-release parameter and a nondimensional activation energy, respectively. The initial conditions are given by

$$T = \exp(x),\ x \leq 0, \tag{11.a}$$

$$Y = 1 - \exp(Le\ x),\ x \leq 0, \tag{11.b}$$

and

$$T = 1,\ x > 0, \tag{11.c}$$

$$Y = 0,\ x > 0, \tag{11.d}$$

the boundary conditions by

$$T = Y - 1 = 0 \text{ as } x \rightarrow -\infty, \tag{11.e}$$

$$\partial T/\partial x = \partial Y/\partial x = 0 \text{ as } x \rightarrow +\infty. \tag{11.f}$$

In all calculations $\alpha = 0.8$ is adopted; β and Le are taken as variable parameters.

As one example, shown in Fig. 1 is the computed propagation velocity v_F of the flame as a function of time for $Le = 1.45$ and $\beta = 32$. It is seen that after a short initial period a continuous limiting cycle results. To accurately resolve the spatial structure of this pulsating flame the adaptive meshing procedure outlined above requires 80 to 100 grid points; to obtain comparable resolution with an equidistant grid as many as 800 to 1000 grid points would be required.w

As a second example we consider auto-ignition in a non-premixed flow generated by directing a hot air stream ($T = 800K$) and a cold fuel stream ($T = 300K$) towards each other.

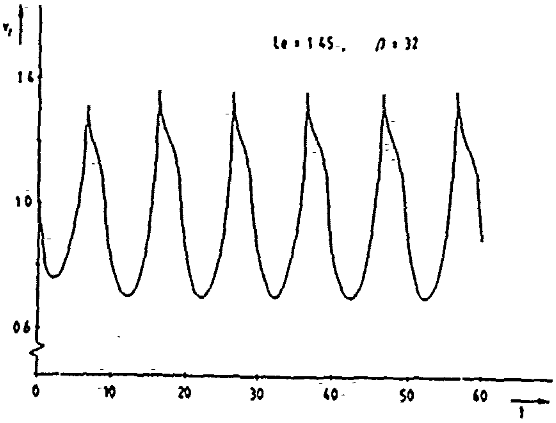


Figure 1: Oscillating flame velocity v_F as a function of time t.

The governing equations are the conservation equations of overall mass, species mass, momentum and energy, space limitations do not allow to present these equations here. Chemistry is assumed to occur via the overall one-step reaction $F + \nu O_2 \longrightarrow P$. For a detailed discussion of appropriate initial and boundary conditions, and the physical background ref. /14/ should be consulted.

As a representative result, Fig. 2 shows temperature profiles during the ignition process. It is seen that a continuous transition takes place from inert mixing at initial time $t = 0$ to a steadily burning diffusion flame. The steadily burning state is reached after roughly 10 microseconds.

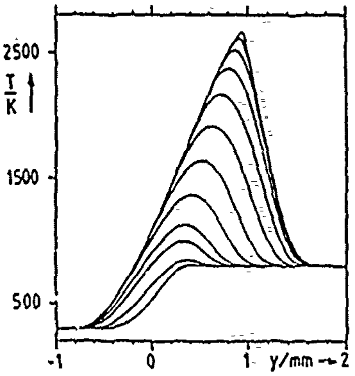


Figure 2: Temperature profiles during a truly time-dependent auto-ignition process. Data: $a = 53.7\ s^{-1}$, $p = 40$ bars, steady-state ignition point at a=58 7s^{-1} The frozen initial profiles pertain to t=0, the other profiles to t=1.01 10^{-9}s, 3.82 10^{-9} s, 6.87 10^{-9}s, 1.57 10^{-8} s, 3.43 10^{-8}s, 7.66 10^{-8} s, 1.71 10^{-7}s, 3.86 10^{-7} s, 9.01 10^{-7}s, 2.25 10^{-6} s, 6.25 10^{-6}s.

As a third example we consider a flame propagating in a strained mixing layer Such flames can be generated, for instance, in a Tsuji like counterflow geometry. The governing equations for this problem were derived by *Liñán* /12/, viz.,

$$u_F \frac{\partial Y_F}{\partial x} - y \frac{\partial Y_F}{\partial y} = \frac{\partial^2 Y_F}{\partial x^2} + \frac{\partial^2 Y_F}{\partial y^2} - \delta \beta^{3/2} Y_F Y_O e^{\beta(T-T_f)/T_f} \tag{11}$$

$$u_F \frac{\partial Z}{\partial x} - y \frac{\partial Z}{\partial y} = \frac{\partial^2 Z}{\partial x^2} + \frac{\partial^2 Z}{\partial y^2}, \tag{12}$$

$$\frac{T - T_o}{T_f - T_o} = 1 - Y_F - Y_O, \tag{13}$$

$$Z = \frac{sY_F + 1 - Y_O}{s + 1}, \tag{14}$$

with the boundary conditions

$$\begin{aligned} & y \rightarrow -\infty : Y_F = Z = 0, \\ & y \rightarrow +\infty : Y_F = Z = 1, \\ & x \rightarrow -\infty : Z = \frac{1}{2} \frac{s+1}{sY_F + 1 - Y_O} erfc(y/\sqrt{2}), \\ & x \rightarrow +\infty : \text{zero } x \text{ gradients for all dependent variables.} \end{aligned} \tag{15}$$

The unknowns in this problem are the (constant) burning-rate eigenvalue u_F, the temperature T, and the mass fractions of fuel and oxidiser, Y_F and Y_O, respectively. The Damköhler number δ and the nondimensional activation energy β are variable parameters to be specified, for the results presented herein we have selected $\delta = 0.5$ and $\beta = 5$.

Shown in Fig. 3 is the computational mesh on which the final solution has been obtained for the above problem. It is seen that the mesh has a tree-like structure due to the generation of individual mesh elements at successive levels $\imath$ of the adaptive meshing procedure. As one example, shown in Fig. 4 is a surface plot of temperature. Notice that mesh points are concentrated in regions where the temperature exhibits steep gradients, and where the surfaces of both temperature and fuel mass fraction have strong curvature.

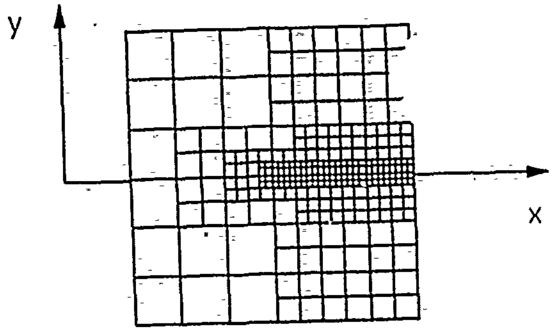


Figure 3: Adaptively generated, converged computational mesh for Example 3.

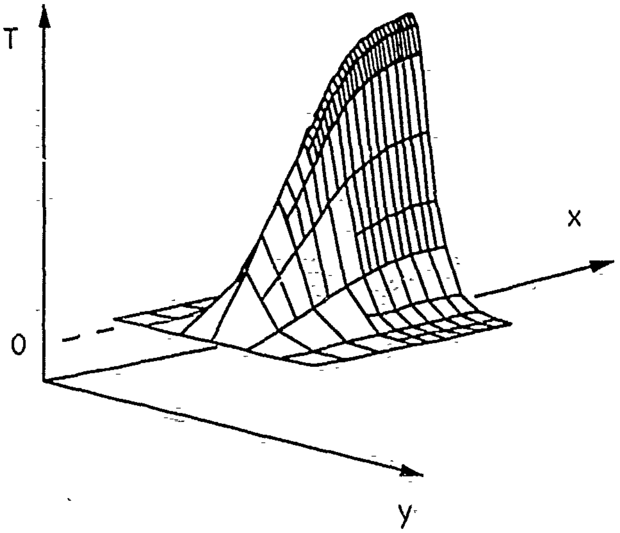


Figure 4: Surface plot of temperature for Example 3.

6. CONCLUSIONS

We have developed and discussed numerical approaches having all the ingredients neccessary for successfully tackling problems typically arising in combustion. Specifically, emphasis has been laid on self-adaptive gridding procedures applicable to time-dependent two-dimensional reactive flows. As examples pulsating flame propagation, auto-ignition in a non-premixed flow, flame propagation in a strained mixing layer and hot-spot-like self-ignition have been considered.

REFERENCES

1. *Dixon-Lewis, G.*: Computer Modeling of Combustion Reactions in Flowing Systems with Transport, in W. C. Gardiner, Jr. (Ed.), Combustion Chemistry, pp. 21-125, Spring New York (1984).
2. Giovangigli, V., Smooke, M.D. Adaptive Continuation Algorithms with Application to Combustion Problems, Report ME-102-87, Yale University, 1987.
3. *Rogg, B.*: Response and Flamelet Structure of Stretched Premixed Methane-Air Flames, Combust. Flame 73, 45-65 (1988).
4. *Benkhaldoun, F., Larrouturou, B.*: Explicit Adaptive Calculations of Wrinkled Flame Propagation, Int. J. for Numerical Methods in Fluids 7, 1147-1158 (1987).
5. *Maas, U., Warnatz, J.*: Numerical Simulation of Ignition Processes, Proc. Joint Meeting of the British and French Sections of the Combustion Institute, Rouen (France), 17-21 April 1989, 133-137 (1989).
6. *Rogg, B*: On Numerical Analysis of Two-Dimensional, Axisymmetric,Laminar Jet Diffusion Flames; in: Mathematical Modeling in Combustion and Related Topics, C.-M. Brauner and C. Schmidt-Lainé (Eds.), Martinus Nijhoff Publishers, 551-560 (1988).
7. *Deuflhard, P.*: A Modified Newton Method for the Solution of Ill-Conditioned Systems of Nonlinear Equations with Application to Multiple Shooting, Numer. Math. 22, pp. 289, 1974.
8. *Smooke, M.D.*: An Error Estimate for the Modified Newton Method with Application to the Solution of 'Ionlinear Two-Point Boundary Value Problems, J. Opt. Theory and Appl. 39, pp. 489 , 1983.
9. *Petzold, L.R.*: A Description of DASSL: A Differential-Algebraic System Solver, Sandia National Labs., Albuquerque, New Mexico, Report SAND82-8637, 1982.
10. *Deuflhard, P., Nowak, U.*: Extrapolation Integrators for Quasilinear Implicit ODEs,Universität Heidelberg, Sonderforschungsbereich 123, Preprint No. 332, 1985.
11. *Deuflhard, P., Hairer, E., Zugck, J.*: One-Step and Extrapolation Methods for Differential-Algebraic Systems, Uni versität Heidelberg, Sonderforschungsbereich 123, Preprint No. 318, 1985.
12. *Fritsch, F.N., Carlson, J.* Monotone Piecewise Cubic Interpolation, SIAM J. Numer. Anal 17, pp 238-246, 1980
13. *Peters, N.*. Discussion of Test Problem A, in Numerical Methods in Laminar Flame Propagation, N. Peters und J. Warnatz (Eds.), Vieweg, Braunschweig/Wiesbaden, 1-14 (1982).
14. *Bruel, P, Rogg, B., Bray, K.N.C.*: On Auto-Ignition in Non-Premixed Laminar and Turbulent Systems, 23nd Symp (Int.) on Comb., The Combustion Institute, Pittsburgh, in press (1990).
15. *Liñán, A.*. Private communication (1989).

A DOMAIN-ADAPTIVE BLOCK BIDIAGONAL TECHNIQUE FOR ANNULAR LIQUID JETS

J. I. Ramos
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890
U.S.A.

Juan Falgueras and Enrique Nava
F. Informática /E.T.S.I. Telecomunicación
Universidad de Málaga
Plaza El Ejido, s/n
29013-Málaga
SPAIN

ABSTRACT - An adaptive, block-bidiagonal finite difference method is used to study the response of annular liquid jets to the injection of mass into the volume enclosed by the annular jet. It is shown that the annular jet's response is characterized by damped oscillations in both the convergence length and the pressure of the gases enclosed by the jet, and that the amplitude and number of these oscillations increase as the initial pressure ratio across the annular jet and the pressure of the gases surrounding the jet are increased.

I INTRODUCTION

Annular liquid jets may form enclosed volumes if the pressure difference between the gases enclosed by and surrounding them is overcome by surface tension, and can be used to measure the dynamic surface tension of liquids and burn toxic wastes in the volume enclosed by the annular jet [1].

The equations which govern the fluid dynamics of inviscid, isothermal, annular liquid jets were derived in Reference [2]. These equations are asymptotic to terms proportional to the annular jet's thickness-to-radius ratio at the nozzle exit and are valid for steady and unsteady jets. In this paper, an adaptive block-bidiagonal finite difference technique is used to study the dynamic response of annular liquid jets to the injection of mass into the volume enclosed by the annular jet as a function of the initial pressure difference between the gases enclosed by and surrounding the jet, and pressure of the gases surrounding the annular jet.

II FLUID DYNAMICS EQUATIONS

The nondimensional equations governing the fluid dynamics of inviscid, isothermal, annular liquid jets can be written as [1]

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial z} = \mathbf{G} \qquad (1)$$

where

$$\mathbf{U} = [m, mR, mu, m\bar{v}]^T, \qquad \mathbf{F} = [mu, mRu, muu, mu\bar{v}]^T \qquad (2)$$

$$\mathbf{G} = \left[0, m\bar{v}, \frac{m}{Fr} + \frac{1}{We}\left(\frac{\partial J}{\partial z} - C_{pn}R\frac{\partial R}{\partial z}\right), \frac{1}{We}\left(C_{pn}R - \frac{\frac{\partial J}{\partial z}}{\frac{\partial R}{\partial z}}\right)\right]^T \qquad (3)$$

$$\mathbf{U}(\tau, 0) = [1, 1, 1, \tan\theta_0]^T \qquad (4)$$

$$Fr = u_0^{*2}/g\,R_0^*, \qquad We = m_0^* u_0^{*2}/2\sigma\,R_0^* \qquad (5)$$

$$C_{pn} = C_p We, \qquad C_p = (p_i^* - p_e^*)\,R_0^{*2}/m_0^* u_0^{*2} \qquad (6)$$

$$J = R\Big/\left[1 + \left(\frac{\partial R}{\partial z}\right)^2\right]^{\frac{1}{2}}. \qquad (7)$$

If the gases enclosed by the annular liquid jet are ideal and isothermal, the liquid does not absorb the gases that it encloses, and mass is injected into the volume enclosed by the jet at a rate a and during a time t_{inj}, then

$$\frac{p_i^*(t)}{p_e^*} = m_i(t)\,\frac{V(0)}{V(t)}\frac{p_i^*(0)}{p_e^*} \qquad (8)$$

$$m_i(t) = \frac{m_i^*(t^*)}{m_i^*(0)} = 1 + at, \qquad 0 \le t \le t_{inj} \qquad (9)$$

$$m_i(t) = 1 + at_{inj}, \qquad t > t_{inj} \qquad (10)$$

Substitution of Eq. (8) into Eq. (6) yields

$$C_{pn} = C_{p\max}\left[m_i(t)\,\frac{V(0)}{V(t)}\frac{p_i^*(0)}{p_e^*} - 1\right] \qquad (11)$$

where

$$C_{p\max} = p_e^* R_0^*/2\sigma, \qquad V(t) = \int_0^{L(t)} R_i^2(t, z)\,dz \qquad (12)$$

$V = V^*/\pi R_0^{*3}$, and $L(t)$ is the axial distance at which the annular jet's inner interface radius is zero, i.e.,

$$P(t, L(t)) = 0 \qquad (13)$$

and

$$b = \frac{m}{R}\frac{b_0^*}{R_0^*}, \qquad R_i = R - b/2 \qquad (14)$$

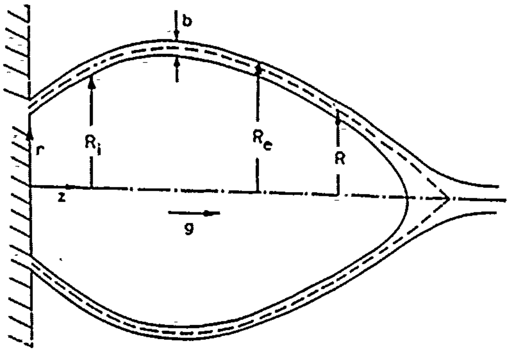


Figure 1. Schematic of an annular liquid jet.

III DOMAIN-ADAPTIVE TECHNIQUE

The annular liquid jet geometry is curvilinear and time-dependent and has an unknown, time-dependent, downstream boundary, i.e., the convergence point. This geometry can be transformed into a unit interval by means of the mapping

$$(t, z) \to (\tau, \eta), \qquad \tau = t, \qquad \eta = z/L \qquad (15)$$

The Jacobian of this mapping is $L(t)$ which is a function of time. Substitution of Eq. (15) into Eq. (1) yields

$$\frac{\partial \mathbf{U}}{\partial \tau} + \mathbf{H}\frac{\partial \mathbf{U}}{\partial \eta} = \mathbf{G}, \qquad \mathbf{H} = \frac{1}{L}\left[\frac{\partial \mathbf{F}}{\partial \mathbf{U}} - \eta\frac{dL}{dt}\mathbf{I}\right] \qquad (16)$$

where $\mathbf{I}$ is the unit matrix. Equation (16) can be discretized in an equally spaced grid such that $\eta_1 = 0$ and $\eta_I = 1$ where I denotes the number of grid points, by means of backward differences for the advection term, and central differences for G, and the resulting $O(\Delta\tau, \Delta\eta)$-accurate finite difference equation can be written as

$$-C_i^{n+1}U_{i-1}^{n+1} + (I + C_i^{n+1})U_i^{n+1} = \Delta\tau\, G_i^{n+1} + U_i^n \tag{17}$$

where $i = 2, 3, \ldots, I-1$, and

$$C = H\,\Delta\tau/\Delta\eta \tag{18}$$

Since both H and C depend on $dL/d\tau$ and L, an equation must be obtained for the convergence length. Such an equation can be obtained as follows. At the convergence point (cf. Eqs. (13) and (14))

$$R_i(t, z = L(t)) = 0, \qquad b(t, z = L(t)) = 2R(t, z = L(t)) \tag{19}$$

Therefore, the following algebraic equation must be satisfied at the convergence point (cf. Eq.(14))

$$2R^2(t, z = L(t)) = b_0^* m(t, z = L(t))/R_0^*. \tag{20}$$

Differentiation of Eq. (20) with respect to t, and use of Eqs. (1) and (15) yield

$$\frac{dL}{dt} = \frac{dL}{d\tau} = \frac{4R(u\frac{\partial R}{\partial \eta} - \overline{v}L) - \frac{b_0^*}{R_0^*}\frac{\partial}{\partial \eta}(mu)}{4R\frac{\partial R}{\partial \eta} - \frac{b_0^*}{R_0^*}\frac{\partial m}{\partial \eta}} \quad \text{at } \eta = 1 \tag{21}$$

which is an ordinary differential equation for L and which can be discretized, in an equally-spaced grid, as

$$\frac{dL}{d\tau} = \frac{4R_I\left(u_I\frac{R_I - R_{I-1}}{\Delta\eta} - \overline{v}_I L\right) - \frac{b_0^*}{R_0^*}\frac{(mu)_I - (mu)_{I-1}}{\Delta\eta}}{4R_I\frac{R_I - R_{I-1}}{\Delta\eta} - \frac{b_0^*}{R_0^*}\frac{m_I - m_{I-1}}{\Delta\eta}}. \tag{22}$$

The values of u, R, $\overline{v}$ and m at $i = I$ can be calculated by linear extrapolation as

$$U_I = 2U_{I-1} - U_{I-2} \tag{23}$$

IV PRESENTATION OF RESULTS

Figures 2 and 3 illustrate the effects of the initial pressure ratio across the annular liquid membrane on both the convergence length and the pressure coefficient. These figures correspond to nonpressurized and overpressurized annular membranes, respectively, and indicate that the initial pressure ratio across the membrane has a great effect on its dynamic response. In particular, Figures 2 and 3 clearly indicate that the time required to reach asymptotic, steady state after the end of mass loading increases as the initial pressure ratio across the membrane is increased, and that overpressurized membranes exhibit damped oscillations analogous to those of a mass-spring-dashpot system.

Figures 2 and 4 illustrate the effects of C_{pmax} on the response of annular membranes subject to mass loading, and indicates that the maximum values of both the pressure coefficient and the convergence length, and the time required to reach asymptotic, steady state after the end of mass injection increase as C_{pmax} is increased. Figure 4 also shows that the critical pressure coefficient of unity determined from the solution of the steady state governing equations [2] can be exceeded without affecting the stability of the annular membrane, and that the initial response of the pressure coefficient is nearly linear.

V CONCLUSIONS

The dynamic response of annular liquid jets to mass loading has been analyzed by means of an adaptive finite difference method that transforms the unknown, time-dependent, curvilinear geometry of the annular jet into a unit interval, and that yields a differential equation for the convergence length, i.e., for the axial distance at which the annular jet merges on the symmetry axis to form a solid jet. A block-bidiagonal technique has been used to determine the annular jet mean radius, mass per unit length, and axial and radial velocity components of the liquid in an iterative manner.

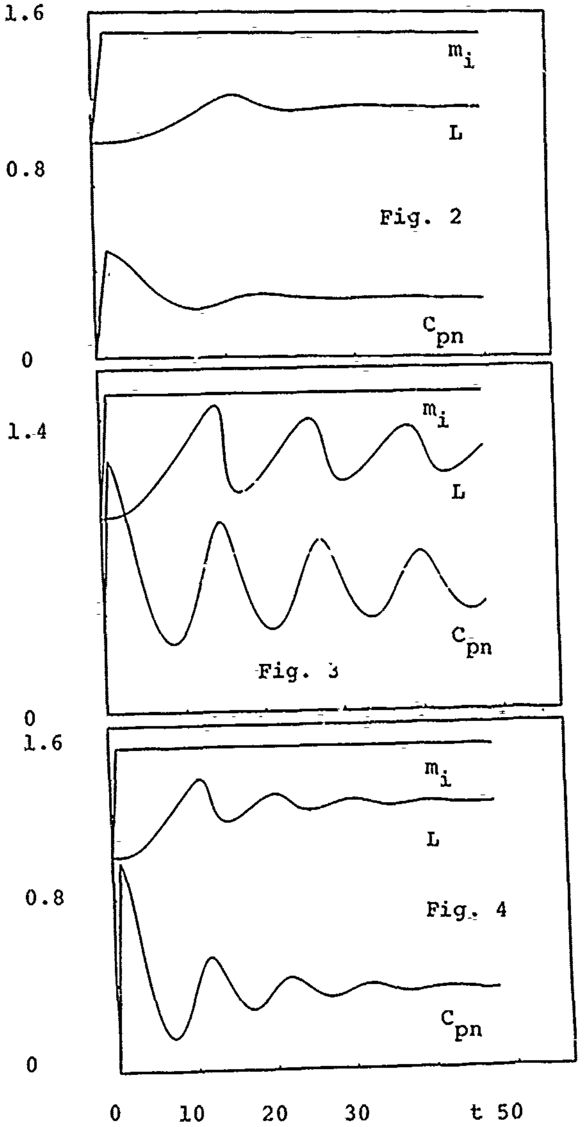


Fig. 2

Fig. 3

Fig. 4

It has been shown that the pressure coefficient and the pressure of the gases enclosed by the annular liquid jet respond instantaneously to the mass injection, whereas there is a lag in the response of the convergence length. This lag is due to the inertia of the jet and the assumption that the gases enclosed by the jet are isothermal, and decreases as the injection duration is increased.

REFERENCES

[1] R. M. Roidt and Z. M. Shapiro, "Liquid curtain reactor", Report No. 85M981, Westinghousec R&D Center, Pittsburgh, Pennsylvania, 1985.

[2] J. I. Ramos, "Annular liquid jets: Formulation and steady state analysis", *Z. angew. Math. Mech. (ZAMM)*, in press (1991).

DIFFUSION CURRENT IN mm-WAVE DDRs

S.P. PATI AND G.N. DASH
Department of Physics, Sambalpur University,Jyoti Vihar, Sambalpur-768019(Orissa) INDIA

ABSTRACT:

Diffusion of charge carrriers would become a prominent physical phenomena in case of short mm-wave DDRs due to narrow depletion zone and high carrier concentration gradient. An economical numerical simu-lation method to compute the diffusion current and the magnitude of diode negative resistance of Si DDRs due to diffusion current density is presented which provides a realistic picture of the effect of carrier diffusion in IMPATT devices for mm-wave operation.

INTRODUCTION:

Ultrathin depletion layer and high mobile space charge concentration in IMPATT diodes operating in mm-wave and short mm-wave regions would push the carrier diffusion current to comparable limit of drift curre-nt and both drift and difusion currents would contr-ibute to microwave negative resistance generated in the device. The inclusion of diffusion current to the mathematical analysis of the device physics is often neglected due to complexities that may be involved in the analysis. The authors have devised a comparative-ly low cost numerical simulation method to solve IMPATT device equations under high frequency small signal conditions by considering both drift and drift currents. Our analysis also can give the microwave resistance contributed by diffusion current only whi-ch in turn can indicate a clear picture as regards role of diffusion current in the device performance. The method has been applied to several mm-wave DDRs designed to operate upto 220 GH_Z. The deteriorating effect of carrier diffusion is marked at frequencies beyond 150 GHz.

DEVICE ANALYSIS:

Inclusion of diffusion current into the analysis of framing the device equations of semiconductor devi-ces, can be realised in a simple way by defining the operators for the effective velocities of electron and hole in the form, $V_{n,p}=v_{n,p}(1+\frac{D_{n,p}}{v_{n,p}}\mathcal{D})$ and sum velocity $V_+ = (v_n+v_p)\,(1+[(D_n-D_p)/(v_n+v_p)]\,\mathcal{D})$ where $v_{n,p}$ are the drift velocities and $D_{n,p}$ are the diffusion coefficients of electrons and holes respect-ively and $D=\partial/\partial x$. The electron and hole current den-sities J_n and J_p now, take the usual form $J_{n,p}=qV_{n,p}(n,p)$ (1)
Where q is the electronic charge and n and p are res-pectively the electron and hole concentrations. The velocity operators have their corresponding inverses.

The basic equations for an IMPATT diode are the combined carrier continuity equation

$$\frac{\partial}{\partial t}(p+n)=\frac{1}{q}\frac{\partial}{\partial x}(J_n-J_p)+2(\alpha_n v_n n+\alpha_p v_p p) \qquad (2)$$

and the Poisson's equation (for movile space charge)

$$(p-n)=\frac{\epsilon}{q}\frac{\partial E_m}{\partial x} \qquad (3)$$

Where $\alpha_{n,p}$ are respectively the ionisation coefficients of electrons and holes.The total current density,which is the sum of conduction current and displacement curr-ent, is constant and is given by

$$J = J_n + J_p + \epsilon\frac{\partial E_m}{\partial t} \qquad (4)$$

J_p and J_n can be eliminated using Equs (1),(3) and (4) to obtain expressions for p and n [1].These expressions for p and n can be substituted into Equs ′ and on simplification one can obtain the following fourth order differential equation,

$$[-D_AD^4+\bar{D}D^3+(1+D_+k-\bar{\alpha}_D)D^2+(\alpha_n-\alpha_p+2rk)\mathcal{D}+2\bar{\alpha}k-k^2]E_m=\frac{1}{\bar{v}\epsilon}(2\bar{\alpha}-k)J \qquad (5)$$

Where the symbols have usual meaning [1].

Equ (5) is linearized to get the small signal device equation on the device impedence Z as

$$[-D_AD^4+\bar{D}D^3+(1+D_+k-\bar{\alpha}_D)\mathcal{D}^2+(\alpha_n-\alpha_p+2rk)D+2\bar{\alpha}k-k^2+(\alpha_n'-\alpha_p')\mathcal{D}E_m-(2\bar{\alpha}'J/\bar{v}\epsilon)-\bar{\alpha}_D'\mathcal{D}^2E_m]Z=(1/\bar{v}\epsilon)(2\bar{\alpha}-k) \qquad (6)$$

Where the primes denote the field derivatives of the corresponding quantities. The boundary conditions are obtained by assuming the electron and hole concentra-tions to be negligible respectively at the p-side and n-side boundaries of the diode. These are given by

$$\epsilon\left(\frac{\partial R}{\partial x}+\frac{\omega X}{v_n}\right)+\frac{1}{v_n}=0\,;\quad \frac{\partial X}{\partial x}-\frac{\omega R}{v_n}=0 \qquad (7)$$

at the n-side boundary and,

$$\epsilon\left(\frac{\partial R}{\partial x}-\frac{\omega X}{v_p}\right)-\frac{1}{v_p}=0\,;\quad \frac{\partial X}{\partial x}+\frac{\omega R}{v_p}=0 \qquad (8)$$

at the p-side boundary.

Assuming that the diffusion current is a small perturbation on the drift component, Equ (6) can be written as

$$(L+\lambda L')\,Z=F \qquad (9)$$

Where L' contains all the terms involving $D_{n,p}$ on the on the L.H.S. of Equ (6) and F is the R.H.S. of Equ(6). Expressing Z as the sum of the unperturbed solution Z_o and various orders of pertubation corrections Z_i(i=1, 2,3....) one can write Equ (9) as

$$(L+\lambda L')\,(Z_0+\lambda Z_1+\lambda^2Z_2+\cdots)=F$$

Equating coefficients of various powers of λ a series of differential equation is obtained as

$$LZ_o = F \qquad (10)$$

$$\text{and}\quad LZ_i = -LZ_{i-1},\quad i = 1, 2, 3 \ldots\ldots \qquad (11)$$

After separating into real and imaginary parts,Equ(10) is first solved and then Equ (11) are solved progress-ively for i = 1, 2, 3 ...... etc. using the numerical technique described as follows.

COMPUTER METHOD:

Equ (10) is solved by a modified Runge-Kutta algo-rithm following an iterative procedure.The iterations over the initial values of resistance R_o (R_eZ_o) and reactance X_o (I_mZ_o) at one edge of the diode are per-formed till the boundary conditions at the other edge are satisfied. A four fold logic is framed in perfor-ming the iterations. The initial values of R_o and X_o may be varied in the four possible ways i.e. R_o $R_o\pm\Delta R_o$ and X_o $X_o\pm\Delta X_o$. The logic in which the initial valu-es of R_o and X_o is to be varied to obtain the required boundary conditions at the other edge depends on the structure of the diode as well as on the frequency of operation. The programme software has been designed in such a way that the variations in the initial values of R_o and X_o automatically switch over to the converg-ing track very swiftly.The software is thus free from numerical instability. The accuracy limit is set at

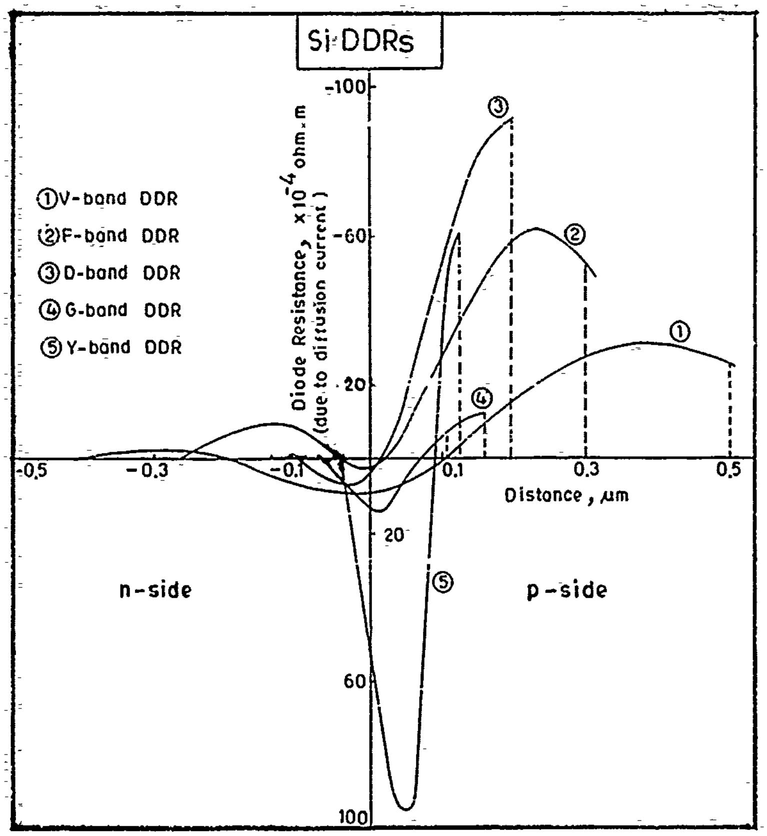


Fig.1 Microwave diode resistance contributed by diffusion current in Silicon DDRs.

0.02% . After the solution of Equ (10) the quantities on R.H.S. of the Equ (11)for i=1 are obtained from the knowledge of R_0 and X_0 at each space point. The third and fourth order derivatives of R_0 and X_0 are obtained numerically following Sterling's formula. Then Equ(11) is solved for i=1 following the same Runge-Kutta approach to get the first order diffusion correction.The different order of diffusion corrections may be obtained by progressively solving Equ (11) for i=2, 3, 4.. etc. which gives the diffusion contribution to the negative resistance of the diode.

RESULTS:

Flat profile Si DDRs for operation in V, F, D, G and Y bands with centre frequencies respectively at 60 94, 140, 170 and 220 GH_Z are designed following a static analysis [2]. The small signal mm-wave properties like diode negative conductance (G) and diode negative resistance Z_R, which are determined following our method are presented in Table-1 for the cases (i) when diffusion is neglected and (ii) when diffusion is considered. It is seen that the effect of carrier diffusion on the device characteristics remains marginal for diodes with frequency of operation below 100 GH_Z. Further it is observed that diffusion current enhances the device negative resistance for diodes designed to operate below 150 GH_Z. The degrading effect of carrier diffusion starts with diodes designed to operate above 150 GH_Z and it substantially reduces the device negative resistance for Y-band operation. The device negative conductance also records a progressive deterioration due to carrier diffusion as one goes from low frequency V-band to high frequency Y-band. The distribution of resistance, contributed by diffusion current, in the depletion layers of different diodes are shown in Fig.1 . The curves show that the diffusion contribution to diode resistance becomes positive for G and Y band diodes giving rise to decrease in the device negative resistance for high frequency mm-wave bands leading to fall in the power output and efficiency of IMPATT devices for short mm-wave operation.

Table-1

Band	Optimum frequency GH_Z	$D_{n,p}=0$		$D_{n,p}\neq 0$	
		$-G$, $\times 10^6 S/m^2$	$-Z_R$, $\times 10^{-9}\Omega m^2$	$-G$, $\times 10^6 S/m^2$	$-Z_R$, $\times 10^{-9}\Omega m^2$
V	60	7.8	7.6	8.1	8.3
F	105	38.1	6.4	37.2	7.6
D	130	56.8	2.3	54.4	3.1
G	170	99.1	1.4	85.6	1.3
Y	220	192	1.0	42.0	0.3

REFERENCES:

1. G.N. Dash and S.P. Pati - Semiconductor Science and Technology, IOP Publishing Ltd., England (To be published).

2. D.N. Datta, S.P. Pati et al, IEEE, ED-29, pp 1813-1816(1982).

EQUATIONS AND NUMERICAL METHODS FOR LINEAR WAVE PROPAGATION IN ANELASTIC MEDIA [*]

JOSE M. CARCIONE
Osservatorio Geofisico Sperimentale,
P. O. Box 2011
34016 Trieste, ITALY.
and
Geophysical Institute, Hamburg University.
Bundesstrasse 55, 2000 Hamburg 13, GERMANY.

AND

ALFRED BEHLE
Geophysical Institute, Hamburg University.
Bundesstrasse 55, 2000 Hamburg 13, GERMANY.

Abstract The equations governing linear wave propagation in viscoelastic media, either single-phase or multiphase, can be written as a single first-order matricial differential equation in time. The formal solution is the evolution operator e^{Mt} acting on the initial condition vector, where M is a linear operator matrix containing the spatial derivatives and medium properties, and t is the time variable. The problem is solved numerically approximating the evolution operator by an optimal polynomial expansion depending on the location of the eigenvalues of M in the complex frequency plane. The eigenvalue analysis is carried out for the anisotropic-viscoelastic and porous viscoacoustic constitutive relations and respective limiting rheologies. For each case an optimal expansion of the evolution operator is identified, which provides highly accurate solutions and fast convergence compared to Taylor expansion or temporal differencing.

I. INTRODUCTION

Linear viscoelasticity provides a general framework for describing the anelastic effects in wave propagation, i.e., the conversion of part of the energy into heat, and the dispersion of the wave field Fourier components with increasing time. A dissipation model which is consistent with real materials is the general standard linear solid which is based on a spectrum of relaxation mechanisms. However, implementation of this rheology in the time domain is not straightforward due to the presence of convolutional kernels (Bolztmann's superposition principle). To avoid the time convolutions, it is necessary to introduce into the formulation additional variables, called memory variables in virtue of their nature [1]- [5] . The wave equation of the medium can be written as a first-order differential equation in time as

$$\dot{U} = MU + F, \tag{1}$$

where U is a vector whose components are the unknown variables, M is an operator matrix containing the spatial derivatives and material properties, and F is the body force vector. In (1) and elsewhere, time differentiation is indicated with the dot convention. The differential equation (1) correctly describes the anelastic effects in wave propagation within the framework of linear response theory.

The solution of (1) subject to the initial condition

$$U(t = 0) = U_0 \tag{2}$$

is formally given by

$$U(t) = e^{tM}U_0 + \int_0^t e^{\tau M} F(t-\tau)\, d\tau. \tag{3}$$

In equation (3), e^{tM} is called the evolution operator of the system. Solving (3) requires a suitable approximation for the spatial derivatives, which is achieved by the Fourier pseudospectral method [7] . Thus, equations (1), (2) and (3) should be replaced by the discretized equivalent equations.
The numerical solution is obtained by an optimal expansion of the evolution operator as polynomials, whose region of convergence depends on the spatial matrix M, particularly on the location of its eigenvalues in the complex frequency plane. The form of M depends on the rheology and the unknown variables.

Let a plane wave solution to equation (1) be of the form

$$U = U_0 e^{i(\omega_c t - k \cdot x)}, \tag{4}$$

where x is the position variable, ω_c is the complex frequency, and k is the real wavenumber vector. Substituting (4) into (1), and considering constant material properties and zero body forces, yields an eigenvalue equation for the eigenvalues $\lambda = i\omega_c$. The determinant of the system must be zero in order for U_0 to have a non-zero value. Therefore,

$$\det[\tilde{M} - \lambda I] = 0, \tag{5}$$

where $\tilde{M}$ is the spatial Fourier transform of M , and I is the identity matrix. Hereafter, the complex plane of the eigenvalues is called the z -plane. Equation (5) determines the eigenvalues of $\tilde{M}$ in the Fourier method approximation. Actually, the discretized equation should be used, but (5) represents a relatively good approximation.

The eigenvalues are analyzed in Section 2 for the following rheologies:

- ANISOTROPIC-VISCOELASTIC
- Isotropic-viscoelastic

[*] This work was supported in part by the Commission of the European Communities under project EOS 1 (Exploration Oriented Seismic Modelling and Inversion), Contract N. JOUF-0033, part of the GEOSCIENCE project within the framework of the JOULE R & D Programme (Section 3.1.1.b).

- Anisotropic-elastic
- Isotropic-elastic
- POROUS ISOTROPIC-VISCOACOUSTIC
- Biot-acoustic
- Isotropic-viscoacoustic

The eigenvalue distribution defines the domain where the evolution operator is approximated by a suitable (rapidly converging) polynomial expansion. For each case, a brief review of the numerical integration techniques is given in Section 3. The methods are the following:

- Taylor expansion
- Chebychev Spectral method
- Rapid expansion method
- Polynomial interpolation through conformal mapping
- Polynomial interpolation by residum minimization

II. WAVE EQUATIONS AND EIGENVALUES OF $\tilde{M}$

Anisotropic-viscoelastic rheology

In order to implement Boltzmann's principle in the generalized Hooke's law, two relaxation functions based on the standard linear solid rheology are considered. One relaxation function describes the anelastic properties of the quasi-dilatational mode, and the other is related to the quasi-shear mode. This can be done by forcing the mean stress to depend on the first relaxation function, and the deviatoric components on the second (in this case, at least for some coordinate system, and usually along symmetry axes of the material). Moreover, the resulting rheological relation gives Hooke's law in the anisotropic-elastic limit, and the isotropic-viscoelastic rheology in the isotropic-anelastic limit [3], [5]. The equation of motion of a two-dimensional anisotropic-viscoelastic medium is formed with the following equations [1]:

i) The equation of momentum conservation:

$$\nabla \cdot T = \rho\ddot{u} + f, \tag{6}$$

where $T^T = [T_1, T_2, T_3, T_4, T_5, T_6] \equiv [\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy}]$ is the stress vector, with σ_{ij}, $i,j = 1,\ldots,3$ the stress components. Defining the position vector by $x = (x, y, z)$, $u(x, t)$ and $f(x, t)$ denote the displacement and body force vectors, respectively; $\rho(x)$ is the density, and $\nabla \cdot$ is a divergence operator defined by

$$\nabla \cdot \rightarrow \nabla_{iJ} = \begin{bmatrix} \partial/\partial x & 0 & 0 & 0 & \partial/\partial z & \partial/\partial y \\ 0 & \partial/\partial y & 0 & \partial/\partial z & 0 & \partial/\partial x \\ 0 & 0 & \partial/\partial z & \partial/\partial y & \partial/\partial x & 0 \end{bmatrix}.$$

ii) The stress-strain relations:

$$T_I = [\Lambda_{IJ} + \Lambda^{(v)}_{IJ} M_{uv}] S_J + \Lambda^{(v)}_{IJ} \sum_{l=1}^{L_v} e^{(v)}_{Jl}, \tag{7}$$

where $I,J = 1,\ldots,6,$ and $v = 1,2.$ $S^T = [S_1, S_2, S_3, S_4, S_5, S_6] \equiv [\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, 2\epsilon_{yz}, 2\epsilon_{xz}, 2\epsilon_{xy}]$ is the strain vector, with ϵ_{ij}, $i,j = 1,\ldots,3$ the strain components; $e^{(v)}_{Jl}$ are memory variables related to the L_v mechanisms which describe the anelastic characteristics of the quasi-dilatational mode ($v = 1$), and quasi-shear modes ($v = 2$); and Λ_{IJ} and $\Lambda^{(v)}_{IJ}$ are functions of the elasticities c_{IJ}, $I,J = 1,\ldots,6$ of the medium. Finally, $M_{uv} = [1 - \sum^{L_v} (1 - \tau^{(v)}_{\epsilon l}/\tau^{(v)}_{\sigma l})]$, where $\tau^{(v)}_{\sigma l}$ and $\tau^{(v)}_{\epsilon l}$ are material relaxation times. Implicit summation over repeated indices is assumed.

ii) the memory variable equations:

$$\dot{e}^{(v)}_{Jl} = S_J \phi_{vl} - e^{(v)}_{Jl}/\tau^{(v)}_{\sigma l}, \qquad l = 1, \ldots, L_v, \tag{8}$$

where $\phi_{vl} = (1 - \tau^{(v)}_{\epsilon l}/\tau^{(v)}_{\sigma l})/\tau^{(v)}_{\sigma l}$.

Equations (6), (7) and (8) are the basis for the numerical solution algorithm. For simplicity, a two-dimensional transversely-isotropic medium with symmetry axis parallel to the z-axis is considered. Then, c_{11}, c_{33}, c_{13} and c_{55} define the elastic characteristics of the medium. Choosing one relaxation mechanism for each mode ($L_1 = L_2 = 1$), the unknown variable vector is given by

$$U^T = [u_x, u_z, \dot{u}_x, \dot{u}_z, e_1, e_2, e_3], \tag{9}$$

where $e_1 = e^{(1)}_{11} + e^{(1)}_{31}$, $e_2 = e^{(2)}_{11} - e^{(2)}_{31}$, and $e_3 = e^{(2)}_{51}$, in terms of the memory variables. The spatial operator is

$$M = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ M_{31} & M_{32} & 0 & 0 & M_{35} & M_{36} & M_{37} \\ M_{41} & M_{42} & 0 & 0 & M_{45} & M_{46} & M_{47} \\ M_{51} & M_{52} & 0 & 0 & M_{55} & 0 & 0 \\ M_{61} & M_{62} & 0 & 0 & 0 & M_{66} & 0 \\ M_{71} & M_{72} & 0 & 0 & 0 & 0 & M_{77} \end{bmatrix}, \tag{10}$$

with

$$\rho M_{31} = \partial/\partial x\ [(c_{11} - D) + (D - c_{55})M_{u1} + c_{55}M_{u2}]\ \partial/\partial x + \partial/\partial z\ (c_{55}M_{u2})\ \partial/\partial z,$$

$$\rho M_{32} = \partial/\partial x\ [(c_{13} + 2c_{55} - D) + (D - c_{55})M_{u1} - c_{55}M_{u2}]\ \partial/\partial z + \partial/\partial z\ (c_{55}M_{u2})\ \partial/\partial x,$$

$$\rho M_{35} = \partial/\partial x\ (D - c_{55}), \quad \rho M_{36} = \partial/\partial x\ c_{55}, \quad \rho M_{37} = \partial/\partial z\ c_{55},$$

$$\rho M_{41} = \partial/\partial z\ [(c_{13} + 2c_{55} - D) + (D - c_{55})M_{u1} - c_{55}M_{u2}]\ \partial/\partial x + \partial/\partial x\ (c_{55}M_{u2})\ \partial/\partial z,$$

$$\rho M_{42} = \partial/\partial z\ [(c_{33} - D) + (D - c_{55})M_{u1} + c_{55}M_{u2}]\ \partial/\partial z + \partial/\partial x\ (c_{55}M_{u2})\ \partial/\partial x,$$

$$\rho M_{45} = \partial/\partial z\ (D - c_{55}), \quad \rho M_{46} = -\partial/\partial z\ c_{55},$$

$$\rho M_{47} = \partial/\partial x\ c_{55},$$

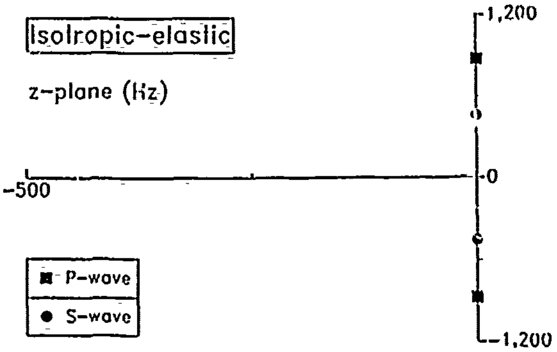
$$M_{51} = \phi_1\ \partial/\partial x, \quad M_{52} = \phi_1\ \partial/\partial z, \quad M_{55} = -1/\tau_\sigma^{(1)},$$

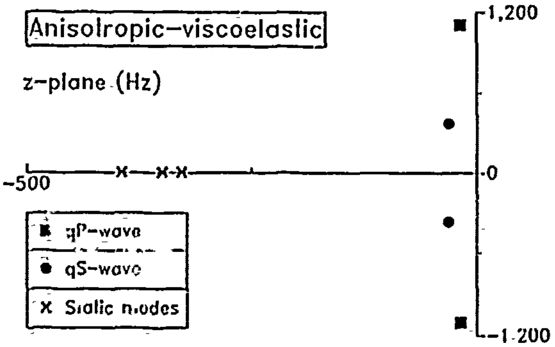
$$M_{61} = \phi_2\ \partial/\partial x, \quad M_{62} = -\phi_2\ \partial/\partial z, \quad M_{66} = -1/\tau_\sigma^{(2)},$$

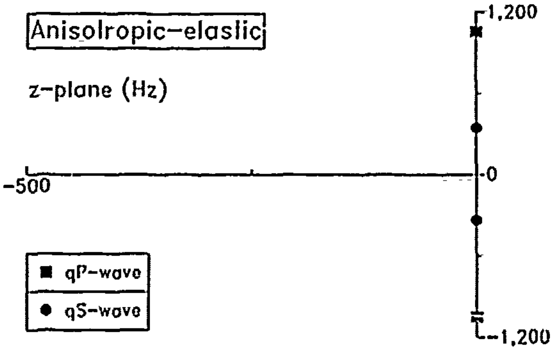
$$M_{71} = \phi_2\ \partial/\partial z, \quad M_{72} = \phi_2\ \partial/\partial x, \quad M_{77} = -1/\tau_\sigma^{(2)},$$

where $D = (c_{11} + c_{33})/2$. The subindex 1 denoting a physical mechanism has been omitted for simplicity. In the anisotropic-elastic limit, i.e., when $\tau_\epsilon^{(\nu)} \rightarrow \tau_\sigma^{(\nu)}$, and the memory variables vanish, equation (2.2) become Hooke's law. In the isotropic-viscoelastic limit, c_{11} , $c_{33} \rightarrow \lambda + 2\mu$, $c_{13} \rightarrow \lambda$ and $c_{55} \rightarrow \mu$, with λ and μ the Lame constants, and (7) becomes the isotropic-viscoelastic rheology [5].

The eigenvalues of $\tilde{M}$ are obtained from equation (5), where the following substitution: $\partial/\partial x \rightarrow ik_x$, and $\partial/\partial z \rightarrow ik_z$, with k_x and k_z the wavenumber components, gives $\tilde{M}$ from M.







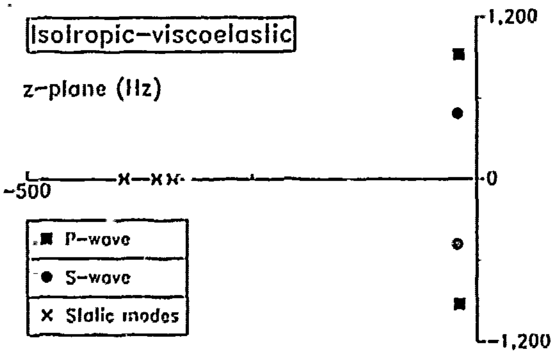


Fig. 1. Eigenvalue distribution of the spatial matrix $\tilde{M}$ in the complex frequency plane for the different rheologies of a single-phase solid.

The eigenvalue distribution for the different rheologies is displayed in Fig. 1. The material is a clayshale having $\tau_\epsilon^{(1)} = \tau_\epsilon^{(2)} = 0.0030\,s^{-1}$, $\tau_\sigma^{(1)} = 0.0027\,s^{-1}$ and $\tau_\sigma^{(2)} = 0.0025\,s^{-1}$, which give highest dissipation around $f = 50$ Hz [1]. The eigenvalues corresponds to $k_x = k_z = 0.16\,m^{-1}$. The negative real part of the propagating modes is a consequence of the anelasticity, stronger for the shear modes. The static modes arise from the fact that the formulation was done in the time domain; they are grouped approximately around $-1/\tau_\sigma^{(1)}$ and $-1/\tau_\sigma^{(2)}$. The differences are mainly due to anelasticity which introduces the static modes, since anisotropy only produces a shift of the wave mode eigenvalues in the vertical direction. Section 3 analyzes the appropriate methods for each rheology.

Porous isotropic-viscoacoustic rheology

Invoking the correspondence principle, Biot formally obtained a viscoelastic equation of motion which includes all possible dissipation mechanisms. The approach involves the presence of convolutional integrals which arise from the replacement of the elastic coefficients by time operators. When standard linear solid kernels are considered for the time operators, the equation of motion of the isotropic-viscoacoustic porous medium is given by the following equations [2]:

i) Biot equations:

$$\nabla \begin{bmatrix} p \\ p_f \end{bmatrix} = \begin{bmatrix} -\rho & \rho_f \\ -\rho_f & m \end{bmatrix} \begin{bmatrix} \ddot{u} \\ -\ddot{w} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \eta/K \end{bmatrix} \begin{bmatrix} \dot{u} \\ -\dot{w} \end{bmatrix} + \begin{bmatrix} s \\ s_f \end{bmatrix}, \tag{11}$$

where p and p_f are the pressure fields of the matrix-fluid system and fluid, respectively; u is the displacement of the solid; w is a vector representing the flow of the fluid relative to the solid, and s and s_f are body force vectors. The material properties are: ρ , the composite density; ρ_s, the solid density; ρ_f , the fluid density; m, the tortuosity; η, the fluid viscosity; and K, the global permeability.

ii) The stress-strain relations:

$$\begin{bmatrix}\sigma\\ p_f\end{bmatrix}=\begin{bmatrix}\psi_1 & \psi_2\\ -\psi_2 & \psi_3\end{bmatrix}\begin{bmatrix}e\\ \zeta\end{bmatrix}+\sum_{l=1}^{L}\left\{\begin{bmatrix}e_{1l}\\ \zeta_{3l}\end{bmatrix}+\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}e_{2l}\\ \zeta_{2l}\end{bmatrix}\right\}, \qquad (12)$$

where e and ζ are the dilatation fields of the solid matrix, and fluid relative to the solid, respectively; and e_{1l}, e_{2l}, ζ_{2l}, and ζ_{3l} are memory variables. $\psi_1 = -(A + R + 2Q)$, $\psi_2 = (Q + R)/\beta$ and $\psi_3 = R/\beta^2$, where A, R, and Q are the classical Biot elastic coefficients, and β is the porosity.

iii) The memory variable equations:

$$\begin{bmatrix}\dot{e}_{1l}\\ \dot{\zeta}_{3l}\end{bmatrix}=\begin{bmatrix}\phi_{1l} & 0\\ 0 & \phi_{3l}\end{bmatrix}\begin{bmatrix}e\\ \zeta\end{bmatrix}-\begin{bmatrix}1/\tau_{\sigma l}^{(1)} & 0\\ 0 & 1/\tau_{\sigma l}^{(3)}\end{bmatrix}\begin{bmatrix}e_{1l}\\ \zeta_{3l}\end{bmatrix}, \qquad (13a)$$

$$\begin{bmatrix}\dot{e}_{2l}\\ \dot{\zeta}_{2l}\end{bmatrix}=\begin{bmatrix}-\phi_{2l} & 0\\ 0 & \phi_{2l}\end{bmatrix}\begin{bmatrix}e\\ \zeta\end{bmatrix}-\frac{1}{\tau_{\sigma l}^{(2)}}\,I\begin{bmatrix}e_{2l}\\ \zeta_{2l}\end{bmatrix}, \qquad (13b)$$

for $l = 1, \ldots, L$, where $\phi_{rl} = -\psi_r L^{-1}(1 - \tau_{\epsilon l}^{(r)}/\tau_{\sigma l}^{(r)})/\tau_{\sigma l}^{(r)}$, $r = 1, 3$, with $\tau_{\sigma l}^{(r)}$ and $\tau_{\epsilon l}^{(r)}$ relaxation times.
In the one-dimensional case with $L = 1$, the unknown vector U has nine components,

$$U^T = [e, \zeta, \dot{e}, \dot{\zeta}, -\dot{w}, e_1, \zeta_3, e_2, \zeta_2]. \qquad (14)$$

The spatial matrix M for constant material properties is given by

$$M = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ M_{31} & M_{32} & 0 & 0 & M_{35} & M_{36} & M_{37} & M_{38} & M_{39}\\ M_{41} & M_{42} & 0 & 0 & M_{45} & M_{46} & M_{47} & M_{48} & M_{49}\\ M_{51} & M_{52} & 0 & 0 & M_{55} & M_{56} & M_{57} & M_{58} & M_{59}\\ M_{61} & 0 & 0 & 0 & 0 & M_{66} & 0 & 0 & 0\\ 0 & M_{72} & 0 & 0 & 0 & 0 & M_{77} & 0 & 0\\ M_{81} & 0 & 0 & 0 & 0 & 0 & 0 & M_{88} & 0\\ 0 & M_{92} & 0 & 0 & 0 & 0 & 0 & 0 & M_{99}\end{bmatrix},$$

(eq. (15)), where

$\gamma M_{31} = [m\psi_1 + \rho_f\psi_2]\Delta$, $\quad \gamma M_{32} = [m\psi_2 - \rho_f\psi_3]\Delta$,

$\gamma M_{35} = (\rho_f\eta/K)\partial/\partial x$,

$\gamma M_{36} = m\Delta$, $\quad \gamma M_{37} = -\rho_f\Delta$, $\quad \gamma M_{38} = -\rho_f\Delta$,

$\gamma M_{39} = m\Delta$,

$\gamma M_{41} = [\rho_f\psi_1 + \rho\psi_2]\Delta$, $\quad \gamma M_{42} = [\rho_f\psi_2 - \rho\psi_3]\Delta$,

$\gamma M_{45} = (\rho\eta/K)\,\partial/\partial x$,

$\gamma M_{46} = \rho_f\Delta$, $\gamma M_{47} = -\rho\Delta$, $\gamma M_{48} = -\rho\Delta$, $\gamma M_{49} = \rho_f\Delta$,

$\gamma M_{51} = [\rho_f\psi_1 + \rho\psi_2]\partial/\partial x$, $\quad \gamma M_{52} = [\rho_f\psi_2 - \rho\psi_3]\partial/\partial x$,

$\gamma M_{55} = \rho\eta/K$,

$\gamma M_{56} = \rho_f\,\partial/\partial x$, $\quad \gamma M_{57} = -\rho\,\partial/\partial x$, $\quad \gamma M_{58} = -\rho\,\partial/\partial x$,

$\gamma M_{59} = \rho_f\,\partial/\partial x$,

$M_{61} = \phi_1$, $\;M_{66} = -1/\tau_\sigma^{(1)}$, $\;M_{72} = \phi_3$, $\;M_{77} = -1/\tau_\sigma^{(3)}$,

$M_{81} = -\phi_2$, $\;M_{88} = -1/\tau_\sigma^{(2)}$, $\;M_{92} = -\phi_2$, $\;M_{99} = -1/\tau_\sigma^{(2)}$,

with $\Delta = \partial^2/\partial x^2$ and $\gamma = \rho_f^2 - \rho m$. Biot poroelastic equations are obtained by taking $\tau_\sigma^{(r)} = \tau_\epsilon^{(r)}$, $r = 1, 3$. Then, the memory variables vanish and the unknown vector becomes $U^T = [e, \zeta, \dot{e}, \dot{\zeta}, -\dot{w}]$. The equation for a viscoacoustic single-phase solid is obtained with $\rho_f = 0$ and $\phi_2 = \phi_3 = 0$; only one set of relaxation times remains, corresponding to the solid phase (ϕ_1). The unknown vector in this case is $U^T = [e, \dot{e}, e_1]$.

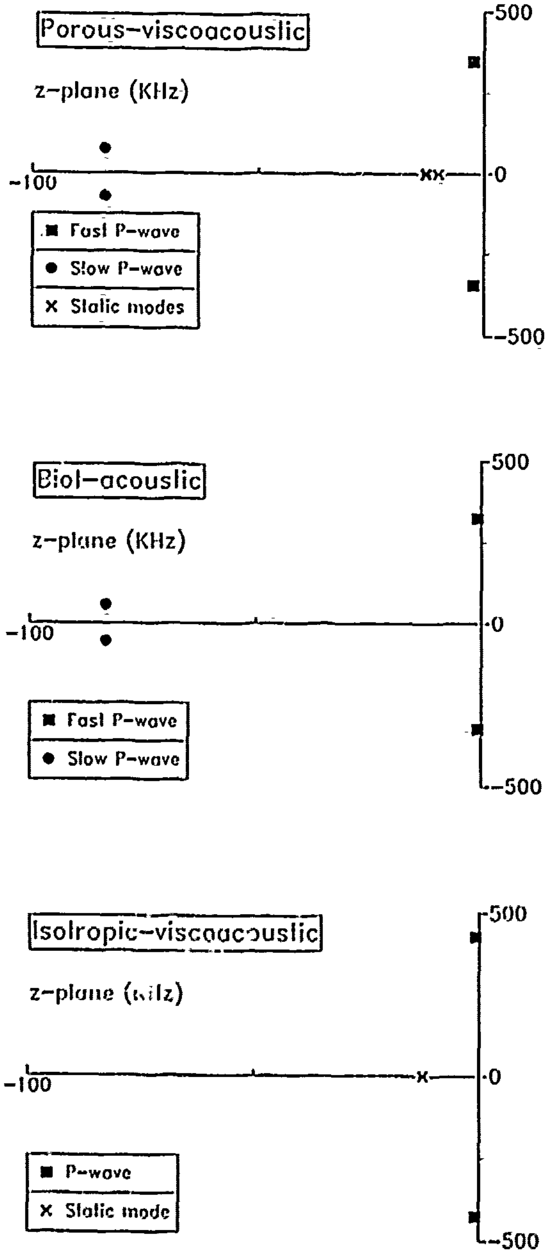


Fig. 2. Eigenvalue distribution of $\tilde{M}$ in the complex frequency plane for a porous viscoacoustic medium and limiting rheologies.

Substitution of $\partial/\partial x$ by ik gives the transformed matrix $\tilde{M}$. Fig. 2 shows the eigenvalue distribution. One of them is zero (not plotted) since the fourth and fifth rows of $\tilde{M}$ are linearly dependent. The slow modes present a quite diffusive behaviour due to the Biot mechanism. They are not present in the single-phase medium, whose attenuation characteristics are viscoelastic.

III. NUMERICAL INTEGRATION METHODS

To illustrate the different techniques, a zero source term is considered for simplicity in equation (1). A detailed formulation with source can be found in the respective references. The formal solution to the system is then given by

$$U(t) = e^{tM}U_0, \tag{16}$$

The numerical solution for general inhomogeneous media requires a polynomial representation of the evolution operator. The different methods are:

<u>Taylor expansion</u> A Taylor expansion of the evolution operator up to the second order is

$$e^{tM} = I + Mt + \frac{1}{2}M^2t^2. \tag{17}$$

Replacing (17) into (16), and substracting $U(-t)$ from $U(t)$ gives

$$U(t) = U(-t) + 2tMU_0. \tag{18}$$

This formula basically gives the equations for second-order temporal differencing valid for small t [7]. Although the region of convergence of the Taylor expansion is the whole z-plane, in order to have high accuracy, the time step should be very small; more precisely, $\Delta t = O(N^{-2})$, using finite-order explicit schemes, where N is the number of grid points.

<u>Chebychev spectral method</u> This technique makes use of the following expansion of e^z [11]:

$$e^z \simeq \sum_{K=0}^{K} C_K J_K(tR) Q_K\left[\frac{z}{tR}\right], \tag{19}$$

where $|z| \leq tR$, and z lies close to the imaginary axis. $C_0 = 1$ and $C_K = 2$ for $k \geq 1$, J_K is the Bessel function of order k , and Q_K are modified Chebychev polynomials which satisfy the recurrence relation

$$Q_{K+1}(s) = 2sQ_K(s) + Q_{K-1}(s), \quad Q_0 = 1, \quad Q_1 - s. \tag{20}$$

Substituting tM for z in (19), equation (16) becomes

$$U(t) \simeq \sum_{K=1}^{K} C_K J_K(tR) Q_K\left[\frac{M}{R}\right] U_0, \tag{21}$$

The series has a rapid convergence for $K > tR$, with $K = O(N)$. The value of R should be chosen larger than the range of the eigenvalues of tM. Since this expansion converges for the imaginary axis of the z-plane, it is appropriate for the elastic case [7]. Anelastic problems can be solved with less efficiency using a slight modification [4].

<u>Rapid expansion method</u> In the elastic case where no first time derivatives of the displacements and memory variables are present, the wave equation of the system can be expressed as

$$\ddot{u} = -L^2u + f, \tag{22}$$

where u is the displacement vector, f is the body force vector, and $-L^2$ is a linear matrix operator similar to M [9]. For zero body forces the formal solution to (22) is

$$u(t) = \cos Lt\, u(0) + \frac{\sin Lt}{L}\dot{u}(0). \tag{23}$$

Adding solutions (23) for times t and $-t$, the displacement time derivative can be eliminated, and the displacement at time t becomes

$$u(t) = -u(-t) + 2\cos(Lt)\, u(0). \tag{24}$$

The method uses the following expansion:

$$\cos Lt \simeq \sum_{K=0}^{K/2} C_{2K} J_{2K}(tR) Q_{2K}\left[\frac{iL}{R}\right], \tag{25}$$

This expansion represents an improvement over the Chebychev spectral method since it contains only even order functions Q_{2K}, however, it can be used only for elastic problems [8].

<u>Polynomial interpolation through conformal mapping</u> As shown in the previous section, in a single-phase anelastic solid, the eigenvalues of M lie on a T-shaped domain D which includes the negative real axis and the imaginary axis. This approach is based on a polynomial interpolation of the exponential function in the complex domain D, on a set of points which is known to have maximal properties. This set, known as Fejer points, is found through a conformal mapping between the unit disc and the domain of the eigenvalues D. In this way, the interpolating polynomial is "almost best" [10].

Getting the Fejer points is as follows: Let $\chi(u)$ be a conformal mapping from the u-plane to z-space, which maps the complement of a disc of radius δ to the complement of D, where δ is the logarithmic capacity of D, given by the limit $\delta = |\chi'(\infty)|$, the prime denoting derivative with respect to the argument. The analytic expression for $\chi(u)$ corresponding to the domain D can be found in [10]. The same function $\chi(u\delta)$ maps the complement of the unit disc to the complement of the domain D.

Then, the Fejer points are $z_j = \chi(u_j)$, $j = 0, \ldots, m-1$ where u_j are the m roots of the equation $u^m = \delta$, with m the degree of the polynomial. The set $[z_j]$, $j = 0, \ldots, m-1$ has maximal properties of convergence. Then, the sequence of polynomials $P_m(z)$ of degree m found by interpolation to an arbitrary function f(z), analytic on D at the points z_j, converge maximally to f(z) on D. The interpolating polynomial in Newton form is

$$P_m(z) = a_0 + a_1(z - z_0) + a_2(z - z_0)(z - z_1) + \ldots$$

$$+ a_m(z - z_0)\ldots(z - z_{m-1}), \tag{26}$$

where $a_j = f[z_0, \ldots, z_j]$, $j = 0, \ldots, m-1$. are the divided differences. The approximating polynomial is given by $P_m(Mt)$ with $f(z)\ e^z$.

Polynomial interpolation by residum minimization The preceding method requires a conformal mapping from the unit disc to the domain of the eigenvalues of M to find the interpolating points. This new technique avoids the conformal mapping by finding the interpolating points automatically in an optimal way [12]. Therefore, the method can be applied for any general matrix M, no matter what the domain D.

The idea is to find the interpolating points by minimizing the L_2 -norm of the error. It is well known that the error of the interpolation is

$$E_m(z) = f(z) - P_m(z) = \frac{f^{(m)}(s)}{m!} R_m, \tag{27a}$$

with

$$R_m(z) = \prod_{i=1}^{m} (z - z_{i-1}) = \sum_{K=0}^{m-1} \alpha_K z^k + z^m \tag{27b}$$

and s the value for which $f(s) - P_m(s) - E_m(s) = 0$. The super-index (m) denotes the $(m)^{th}$ derivative. Substituting Mt for z in (27a) and using (16), the error of the algorithm is

$$E_m = \frac{f^{(m)}(s)}{m!} \Sigma_m, \quad \text{where } \Sigma_m = R_m(Mt)U_0. \tag{28a - b}$$

Minimizing the L_2-norm $\|\Sigma_m\|^2 = (\Sigma_m, \Sigma_m)$ is achieved by solving the following set of m linear equations:

$$\frac{\partial}{\partial \alpha_i} \|\Sigma_m\|^2 = 0, \qquad i = 0, \ldots\ldots, m-1. \tag{29}$$

This is equivalent to solve the following system:

$$DA = B, \qquad A = [\alpha_0, \ \ldots \alpha_{m-1}]^T, \tag{30}$$

where

$$D_{ij} = \left((Mt)^{i-1}U_0, \ (Mt)^{j-1}U_0\right), \tag{31a}$$

$$B_i = -\left((Mt)^{i-1}U_0, \ (Mt)^m U_0\right), \qquad 1 \leq i,j \leq m. \tag{31b}$$

After solving for A, the interpolating points are obtained from the roots of $R_m(z)$. The approximating polynomial is given by (26) with $f(z) = e^z$. Further research is required to determine whether this technique improves the efficiency when solving anelastic wave propagation problems.

IV. CONCLUSIONS

This work briefly reviews some of the theories and algorithms for solving wave propagation problems in linear viscoelastic media. The methods use spectral techniques and solve the wave equation in the time-domain. A consistent introduction of Boltzmann's after-effect principle in the time-domain, for anisotropic and porous media, is achieved by the introduction of memory variables. Some additional assumptions are required in the anisotropic case for the determination of the constitutive relations. The eigenvalue analysis for each rheology indicates that spectral Chebychev methods are suitable for elastic problems, and that polynomial interpolation techniques are required when the medium is anelastic.

REFERENCES

[1] J. M. CARCIONE, Wave propagation simulation in anisotropic linear viscoelastic media: theory and simulated wavefields, Geophys. J. Int., 101, pp. 739-750, 1990.

[2] J. M. CARCIONE, A wave equation for viscoacoustic porous media, submitted to Wave Motion, 1990.

[3] J. M. CARCIONE AND A. BEHLE, Two dimensional and three-dimensional forward modeling in isotropic-viscoelastic media, SEG abstracts, 59th Annual Meeting, 2, pp. 1050-1052, 1989.

[4] J. M. CARCIONE, D. KOSLOFF AND R. KOSLOFF, Wave propagation simulation in a linear viscoacoustic medium, Geophys. J. Roy. Astr. Soc., 93, pp. 621-638, 1988. Erratum: 95, pp. 642, 1988.

[5] __________, Wave propagation simulation in a linear viscoelastic medium, Geophys. J. Roy. Astr. Soc., 95, pp. 621-638, 1988.

[6] __________, Wave propagation simulation in an elastic anisotropic (transversely isotropic) solid, Q. Jl. Mech. Appl. Math., 41, pp. 319-345, 1988.

[7] D. GOTTLIEB AND S. ORSZAG, Numerical Analysis of Spectral Methods, Theory and Applications, CBMS-NSF Regional Conference Series in Applied Mathematics 26, Society for Industrial and Applied Mathematics, Philadelphia, 1977.

[8] D. KOSLOFF, J. M. CARCIONE, B. ROMMEL AND A. BEHLE, Three-dimensional wave propagation simulation in elastic-anisotropic media, SEG abstracts, 59th Annual Meeting, 2, pp. 1016-1018, 1989.

[9] D. KOSLOFF, A. QUEIROZ FILHO, E. TESSMER AND A. BEHLE, Numerical solution of the acoustic and elastic wave equations by a new rapid expansion method, Geophys. Prosp., 37, pp. 383-394, 1989.

[10] H. TAL-EZER, J. M. CARCIONE AND D. KOSLOFF, An accurate and efficient scheme for wave propagation in linear viscoelastic media, Geophysics, 55, pp. 1366-1379, 1990.

[11] H. TAL-EZER, Spectral methods in time for hyperbolic equations, SIAM J. Numer. Anal., 23, pp. 11-26, 1986.

[12] H. TAL EZER, Polynomial approximation of functions of matrices by residum minimization, Personal communication, 1990.

ON THE SOLUTION OF SOME HEAT TRANSFER PROBLEMS WITH JUMPS IN FLUXES ARISING FROM BUILDING PHYSICS

J. KAČUR
Institute Numerical Analysis
Comenius University
84 215 Bratislava, CSR

R. VAN KEER
Seminar Mathematical Analysis
State University Gent
9000 Gent, BELGIUM

Abstract - This paper deals with a new method for some heat transfer problems through a system of walls and caves in buildings, the caves being ventilated and heated. From an energy balance argument in the caves, these problems may be reduced to heat transfer problems through multicomponent media with jumps in both the fluxes and the temperatures at the interfaces of the subregions. The ventilation and heating lead to non standard transition conditions involving Volterra operators, acting on the traces of the temperature from both sides of the interfaces.

Crucial in the analysis is a non standard variational formulation of the problem, taking into account the non perfect thermal contact conditions in an appropriate way. By the method of discretization in time, see e.g. [1], the existence of a unique, stable weak solution may be shown. The resulting recurrent system of elliptic problems at each subsequent time point is approximated by a FEM. Both the convergence and the error estimates for the semi-discrete and fully discrete approximation scheme are stated. In a simple case the numerical results show a good agreement between the exact and the approximate solution.

1. INTRODUCTION.

To avoid distracting technicalities, we confine ourselves to a 1D-model problem with practical relevance viz. the simple cavity structure of Fig. 1., see e.g. [4]

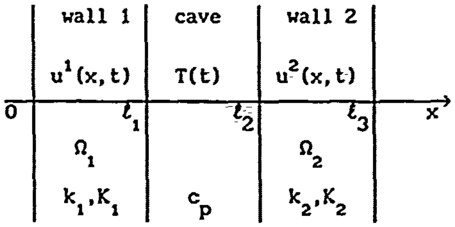


Fig. 1. Cross section of a simple cavity structure

The air space, assumed to be homogeneously at temperature T(t), is contained between two walls (with conductivities k^i and diffusivities K^i), being at temperatures u^i, $i = 1,2$. At $x = \ell_1$ and $x = \ell_2$ heat is transferred by convection between the wall surfaces and the air, with respective transmission coefficients $h^{1,2}$ and $h^{2,1}$. The radiative heat transfer between these surfaces is linearized, with coefficients $H^{1,2} = H^{2,1}$.

The walls consist of two parallel isotropic slabs, assumed to conduct heat only in one direction, orthogonal to the surfaces. As we focus on the contact problem, we take the surfaces $x = 0$ and $x = \ell_3$ to be insulated, for simplicity in the formulation. However inhomogeneous Neumann and Dirichlet or Robin conditions can be covered as well by the present approach.

The cave is heated homogeneously at a rate q(t). Finally the cave is ventilated, both directly by incoming air with velocity v(t) and temperature $T_0(t)$, and indirectly by a change of air with rate K per unit time interval per unit temperature difference, the outdoor air temperature being θ(t). In modelling the ventilation we assume that :

- the incoming air is mixed very quickly with the air in the cave
- the mass of the incoming air equals to the mass of the outgoing air at every time
- the temperature and convective transfer are uniform over the wall surfaces $x = \ell_1$ and $x = \ell_2$.

2. MATHEMATICAL MODEL

Under standard assumptions the mathematical problem of the heat transfer through the cavity structure is : determine $u^i(x,t)$, $x \in \Omega_i$, $t > 0$, $1 \le i \le 2$, or either $T(t)$, $t > 0$, which obey the respective heat equations, with $w^i = k^i/K^i$,

$$w^i(x,t)\cdot \frac{\partial u^i}{\partial t} - \frac{\partial}{\partial x}\left(k^i(x,t)\cdot\frac{\partial u^i}{\partial x}\right) = 0, \quad x \in \Omega_i, \ t > 0, \tag{2.1}$$

together with the boundary conditions

$$\frac{\partial u^1(0,t)}{\partial x} = 0 \ , \ t > 0 \ ; \ \frac{\partial u^2(\ell_3,t)}{\partial x} = 0 \ , \ t > 0 \tag{2.2}$$

as well as with the transition conditions

$$- k^1\cdot\frac{\partial u^1(\ell_1,t)}{\partial x} = h^{1,2}\cdot(u^1(\ell_1,t) - T(t)) + H^{1,2}\cdot(u^1(\ell_1,t) - u^2(\ell_2,t)), \ t > 0 \tag{2.3}$$

$$k^2\cdot\frac{\partial u^2(\ell_2,t)}{\partial x} = h^{2,1}\cdot(u^2(\ell_2,t) - T(t)) + H^{2,1}\cdot(u^2(\ell_2,t) - u^1(\ell_1,t)), \ t > 0 \tag{2.4}$$

and the initial conditions

$$u^i(x,0) = u^i_0(x) \qquad x \in \Omega_i \ , \ t > 0 \tag{2.5}$$

Here, expressing the heat energy balance in the cave, the tmperature T(t) of the inside air is readily seen to evolve from the initial value T(0) according to

$$c_p\cdot \frac{dT(t)}{dt} = g_1\cdot[h^{1,2}\cdot(u^1(\ell_1,t) - T(t)) + h^{2,1}\cdot(u^2(\ell_2,t) - T(t))] - K\cdot[T(t) - \theta(t)] + g_2\cdot v(t)\cdot c_p[T_0(t) - T(t)] + q(t), \ t > 0 \tag{2.6}$$

where

c_p = thermal capacity of the air in the volume V between two cross sections of the cave

$g_1 = S_1/V$, where S_1 is the area of the wall surfaces between these two cross sections

$g_2 = S_2/V$, where S_2 is the area of a cross section of the cave.

Consequently,

$$T(t) = e^{-\kappa(t)}\cdot T(0) + \frac{1}{c_p}\cdot e^{-\kappa(t)}\cdot\int_0^t e^{\kappa(s)}\cdot \{g_1\cdot[h^{1,2}\cdot u^1(\ell_1,s) + h^{2,1}\cdot u^2(\ell_2,s)] + K\cdot\theta(s) + g_2\cdot c_p\cdot v(s)\cdot T_0(s) + q(s)\}\cdot ds \equiv G(u^1,u^2)(t) \tag{2.7}$$

where

$$\kappa(t) = \int_0^t [(g_1/c_p)\cdot(h^{1,2} + h^{2,1}) + g_2\cdot v + K/c_p]\cdot ds.$$

Substituting this expression in the transition conditions (2.3,-(2.4) and translating the interval (ℓ_2,ℓ_3) to (ℓ_1,L), with $L = \ell_1 + \ell_3 - \ell_2$, we arrive at a parabolic problem for the functions $u^i(x,t)$, $x \in \Omega_i$, $t > 0$, $i = 1,2$, of the type mentioned above. This is a problem in a two-component medium with a jump both in the flux and in the temperature at the interface $x = \ell_1$ of the 2 subregions.

Remark 2.1. In practice, when T(t) varies slowly in time, the left hand side of (2.6) may be neglected, c_p being small (in popular models, even $c_p = 0$, [4]). Then $T(t) \equiv G(u^1,u^2)(t)$ takes a particular simple form.

The analysis may easily be extended to a structure with M parallel walls and M-1 enclosed caves (M > 2).

3. VARIATIONAL FORMULATION

3.1. Notations and assumptions

Let $H^1(\Omega_i)$ be the usual first order Sobolev space on Ω_i, with norm $\|.\|_i$, $1 \le i \le 2$. We set

$$V = \{u = (u^1,u^2) | u^i \in H^1(\Omega_i),\ 1 \le i \le 2\}$$

and we identify $u \in V$ with the scalar function $u : \Omega \to R$ for which $u|_{\Omega_i} = u^i$ on Ω^i, $1 \le i \le 2$. Similarly we deal with the product space $H = L_2(\Omega_1) \times L_2(\Omega_2)$. Denote

$$(u,v)_H = \sum_{i=1}^{2} \int_{\Omega_i} u^i\cdot v^i\cdot dx\ ; \quad b(t;u,v) = \sum_{i=1}^{2} \int_{\Omega_i} w^i\cdot u^i\cdot v^i\cdot dx$$

$$|u|_H = (u,u)^{1/2} \qquad \forall u,v \in H \qquad (3.1)$$

$$a(t;u,v) = \sum_{i=1}^{2} \int_{\Omega_i} k^i\cdot\frac{\partial u^i}{\partial x}\cdot\frac{dv^i}{dx}\cdot dx\ , \quad \|v\| = [\sum_{i=1}^{2} \|v^i\|_i^2]^{1/2}$$

$$\forall u,v \in V$$

Let T > 0 be a given number and set I = (0,T). We use the standard functional spaces C(I,X), $L_2(I,X)$, $L_\infty(I,X)$, etc., where X is a Banach space.

Throughout we make the assumptions

$$w^i,k^i \in Lip(I,L_\infty(\Omega_i)); \quad w^i,k^i \ge p > 0 \text{ in } \Omega_i \times I$$
$$(p \text{ constant}),\ 1 \le i \le 2$$

$$h^{1,2},\ h^{2,1} \text{ and } H^{1,2} = H^{2,1} \in Lip(I,R^+)$$

$$v(t),\ \theta(t),\ T_0(t),\ q(t) \in Lip(I,R)$$

$$u_0^i \in H^1(\Omega_i)\ , \quad 1 \le i \le 2.$$

3.2. Variational problem

Definition 3.1. A function $u : I \to H$, with $u \in L_\infty(I,V)$ and $\partial_t u \in L_2(I,H)$, is a variational solution of (2.1)-(2.5), (2.7), in the time interval I, iff

$$\begin{aligned} & b(t;\ \partial_t u(t),\varphi) + a(t;\ u(t),\varphi) \\ & \quad + h^{1,2}\cdot[u^1(\ell_1,t) - G(u^1,u^2)(t)]\cdot\varphi^1(\ell_1) \\ & \quad + h^{2,1}\cdot[u^2(\ell_1,t) - G(u^1,u^2)(t)]\cdot\varphi^2(\ell_1) \\ & \quad + H^{1,2}\cdot[u^1(\ell_1,t) - u^2(\ell_1,t)]\cdot[\varphi^1(\ell_1) - \varphi^2(\ell_1)] = 0 \\ & \qquad \forall\varphi \in V\ ,\ \text{a.e. in } I \end{aligned} \qquad (3.2)$$

$$u(0) = u_0 \qquad (3.3)$$

From the smoothness of the data and the required regularity of u, all integrals in (3.2) exist.

The relation (3.2) is obtained from (2.1)-(2.4) by first dealing with the problems for u^1 and u^2 in the usual way and by next adding the resulting variational equations, taking into account the notations (3.1). The formal equivalence of the classical and the weak variational problem can readily be proved.

4. DISCRETIZATION IN TIME

Consider $n \in N$, a time step $\Delta t = T/n$ and time points $t_j = j\cdot\Delta t$, $0 \le j \le n$. We define $u_j \in V$, intended to be an approximation of $u(x,t_j)$, $1 \le j \le n$, by the linear recurrent system

$$\begin{aligned} & b(t_j;\ \delta u_j,\varphi) + a(t_j;\ u_j,\varphi) \\ & \quad + h_j^{1,2}\cdot[u_j^1(\ell_1) - G(\tilde{u}_{j-1}^1,\tilde{u}_{j-1}^2)(t_j)]\cdot\varphi^1(\ell_1) \\ & \quad + h_j^{2,1}\cdot[u_j^2(\ell_1) - G(\tilde{u}_{j-1}^1,\tilde{u}_{j-1}^2)(t_j)]\cdot\varphi^2(\ell_1) \\ & \quad + H_j^{1,2}\cdot[u_j^1(\ell_1) - u_j^2(\ell_2)]\cdot[\varphi^1(\ell_1) - \varphi^2(\ell_1)] = 0 \\ & \qquad \forall\varphi \in V,\quad 1 \le j \le n \end{aligned} \qquad (4.1)$$

where

$$\delta u_j = (u_j - u_{j-1})/\Delta t\ ,\quad h_j^{1,2} = h^{1,2}(t_j),\ \text{etc.}$$

$$\tilde{u}_{j-1}(t) = \begin{cases} u_r \text{ for } t \in (t_{r-1},t_r),\ 1 \le r \le j-1 \\ u_{j-1} \text{ for } t \in (t_{j-1},T) \end{cases}$$

By the Lax-Milgram lemma it can be shown that the (elliptic) problem (4.1) for $u_j \in V$ has a unique solution in terms of $u_0,\dots,u_{j-1}$, $1 \le j \le n$.

Definition 4.1. The Rothe function $u^{(n)} : I \to V$, intended to be an approximation of u, is introduced by

$$u^{(n)}(t) = u_{j-1} + \delta u_j\cdot(t - t_{j-1}),\ t_{j-1} \le t \le t_j,\ 1 \le j \le n. \qquad (4.2)$$

From here on C > 0 is a generic constant neither depending on Δt nor (in § 5) on λ.

Proceeding to some extent similarly as in [2]-[3], we may prove

THEOREM 4.1. There exists a function $u \in C(I,H) \cap L_2(I,V)$, with $\partial_t u \in L_2(I,H)$ (u is differentiable a.e. in I) such that $u^{(n)} \to u$ in $C(I,H) \cap L_2(I,V)$ and $\partial_t u^{(n)} \to \partial_t u$ in $L_2(I,H)$ for $n \to \infty$. Moreover

$$\|u - u^{(n)}\|_{C(I,H)} + \|u - u^{(n)}\|_{L_2(I,V)} \le C\cdot 1/\sqrt{n} \qquad (4.3)$$

Finally u is the solution in the sense of Defin. 1.1.

Remark 4.1. When the inital function $u_0 \in V$ satisfies a 'compatibility condition', viz. when there exists $z_0 \in H$, to be interpreted as δu_0, such that (4.1) holds for j = 0 too, then the estimate (4.3' can be improved to O(1/n). Moreover, then $u \in Lip_{1/2}(I,V)$ and $\partial_t u \in L_2(I,V) \cap L_\infty(I,H)$.

5. FULL DISCRETIZATION

5.1. Abstract error estimate

Consider a family $(V_\lambda)_{\lambda\to 0}$ of (finite element) subspaces of V. Introduce $u_j^\lambda \in V_\lambda$, the Galerkin approximation of u_j, $0 \le j \le n$, by a similar recurrent system to (4.1), where now $\varphi \in V_\lambda$ and where u_0 is replaced by a suitably choosen approximation $u_0^\lambda \in V_\lambda$.

Put $\alpha = (\Delta t, \lambda)$. The discrete Rothe function $u^{(\alpha)}$ is defined by a similar relation to (4.2). To obtain the counterpart of Theorem 4.1, we assume

$$u_0^\lambda \to u_0 \text{ in } V \quad \text{if} \quad \lambda \to 0 \qquad (5.1)$$

For any $z \in L_2(I,V)$ there exists $z_\lambda \in L_2(I,V_\lambda)$ such that $z_\lambda \to z$ in $L_2(I,V)$ if $\lambda \to 0$ (5.2)

For finite element choices of $V_\lambda \subset V$, (5.2) is implied by an inequality of the type (5.5) below.

THEOREM 5.1. Under the additional assumptions (5.1)-(5.2) the convergences, mentioned in Theorem 4.1, are valid for $u^{(\alpha)}$ if $\alpha \to 0$. Moreover,

$$\|u^{(\alpha)} - u\|^2_{C(I,H)} + \|u^{(\alpha)} - u\|^2_{L_2(I,V)}$$

$$\leq C\cdot[\Delta t + |u_0 - u_0^\lambda|^2_H + \|u - w\|_{L_2(I,H)} + \|u - w\|^2_{L_2(I,V)}$$

$$\forall w \in L_2(I,V_\lambda) \qquad (5.3)$$

Remark 5.1. This estimate can be improved, viz. Δt may be replaced by $(\Delta t)^2$, when u_0^λ satisfies a 'compatibility relation'. More precisely $z_0^\lambda \in V_\lambda$, defined by a similar relation to (4.1) with $j = 0$, $\varphi \in V_\lambda$, and u_0 replaced by u_0^λ, has to obey $|z_0^\lambda|_H < C$ if $\lambda \to 0$.

5.2. Rate of convergence in the mesh parameter

Let $(\tau_\lambda^i)_{\lambda\to 0}$ be a regular family of partitions of Ω_i, $1 \leq i \leq 2$, with global mesh parameter λ. Introduce

$$X_\lambda^i = \{ v^i \in C^0(\bar{\Omega}_i) \mid v^i_{|K} \in P_1(K) \qquad \forall K \in \tau_\lambda^i \}$$
$$V_\lambda = \{ v = (v^1,v^2) \mid v^i \in X_\lambda^i , \quad 1 \leq i \leq 2 \} \subset V \qquad (5.4)$$

For the Lagrange finite element spaces X_λ^i we know

$$|v^i - v^i_\lambda|_{L_2(\Omega_i)} + \lambda\cdot\|v^i - v^i_\lambda\|_1 \leq C\cdot\lambda^2\cdot\|v^i\|_{H^2(\Omega_i)}$$

$$\forall v^i \in H^2(\Omega_i) \qquad (5.5)$$

where $v^i_\lambda = \pi^i_\lambda v^i$ and $\pi^i_\lambda : C^0(\bar{\Omega}_i) \to X^i_\lambda$ is the standard Lagrange interpolator or it's Clement's generalization to $L_2(\Omega_i)$, $1 \leq i \leq 2$.

Moreover, taking $u_0^\lambda = (\pi^1_\lambda u_0^1, \pi^2_\lambda u_0^2)$, we get

$$|u_0 - u_0^\lambda|_H \leq C\cdot\lambda\cdot\|u_0\|.$$

Combining these estimates with Theorem 5.1, we have

THEOREM 5.2. Assume that $u^i \in L_2(I,H^2(\Omega_i))$, $1 \leq i \leq 2$. Take $V_\lambda \subset V$ and $u_0^\lambda \in V_\lambda$ as indicated above, then

$$\|u^{(\alpha)} - u\|^2_{C(I,H)} + \|u^{(\alpha)} - u\|^2_{L_2(I,V)} = O(\Delta t + \lambda^2)$$

If the compatibility condition on u_0^λ, mentioned in Remark 5.1, is satisfied, then this estimate is improved to $O((\Delta t)^2 + \lambda^2)$.

6. NUMERICAL EXAMPLE

We consider a simple test problem, cfr. Remark 2.1, the exact solution of which is known :

Determine u^1 in (0,1) and u^2 in (1,2), $t > 0$, obeying the D.E.'s (2.1), where $k^1 = 1$, $k^2 = 2$, $w^1 = w^2 = 1$, together with the B.C.'s (2.2) at $x = 0$ and $x = 2$ respectively, as well as with the T.C.'s at $x = 1$

$$-\partial_x u^1 = 5\cdot(u^1 - u^2) + 5, \quad 2\cdot\partial_x u^2 = 5\cdot(u^2 - u^1) - 3 ,$$

and the I.C.'s

$$u^1(x,0) = \cos(\beta_1\cdot x),$$

$$u^2(x,0) = (-1/\sqrt{2})\cdot\sin(\beta_1)/\sin(\beta_1/\sqrt{2})\cdot\cos((\beta_1/\sqrt{2})\cdot(x-2))$$

Here $\beta_1 = 1.590724$ is the first positive root of

$$\text{cotg } \beta + (1/\sqrt{2})\cdot\text{cotg}(\beta/\sqrt{2}) = \beta/5$$

By the method of seperation of variables the analytical solution reads

$$u^1(x,t) = e^{-\beta_1^2\cdot t}\cdot u_0^1(x), \quad u^2(x,t) = e^{-\beta_1^2\cdot t}\cdot u_0^2(x) + x$$

As in § 5.2 we use a linear FEM with nodes

$$x^i_\ell = \delta_{i-1,1} + \ell\cdot\lambda, \quad 0 \leq \ell \leq \frac{1}{\lambda} \in \mathbb{N}_0, \quad 1 \leq i \leq 2$$

As error characteristics we use discrete L_1- and L_∞-norms

$$e_1(t) = \sum_{i=1}^{2} \sum_{\ell=0}^{1/\lambda} |u(x^i_\ell,t) - u^{(\alpha)}(x^i_\ell,t)|\cdot\lambda$$

$$e_1(t)\% = 100\cdot e_1(t) / \sum_{i=1}^{2} \sum_{\ell=0}^{1/\lambda} |u(x^i_\ell,t)|\cdot\lambda$$

$$M(t) = \max_i \max_\ell |u(x^i_\ell,t) - u^{(\alpha)}(x^i_\ell,t)|$$

The table below shows accurate results, even for a coarse mesh.

t		0.1	0.2	0.3	0.4	0.5
$10^2 \cdot e_1(t)$	I	0.04	0.09	0.14	0.19	0.2
	II	0.002	0.0038	0.0049	0.005	0.006
$10^2\cdot e_1(t)\%$	I	6.9	19	40	70	112
	II	0.449	0.993	1.65	2.45	3.43
$10^2\cdot M(t)$	I	0.09	0.15	0.18	0.2	0.2
	II	0.001	0.0025	0.003	0.0034	0.0036

Table. Case I : $\Delta t = 10^{-2}$, $\lambda = 1/4$.
Case II : $\Delta t = 10^{-4}$, $\lambda = 1/32$.

Using a Crank-Nicholson modification of the semidicrete scheme of § 4, the results may be improved by about 30 %. This and other generalizations, e.g. to 3D problems including non-linear phenomena, are investigated in a forthcoming paper.

REFERENCES

[1] Kačur J., *Method of Rothe in Evolution equations*, Teubner, Leipzig (1985).

[2] Kačur J., *Application of Rothe's method to integro-differential equations*, J. Reine Angew. Math., 338, 73-105 (1988).

[3] Kačur J., Van Keer R., *On a Rothe-Galerkin finite element method for a parabolic problem with a Volterra operator in the boundary condition*, in : Whiteman J.R., The Mathematics of Finite Elements and its applications, Academic Press, London, (1991)(to appear).

[4] Pratt A.W., *Heat Transmission in Buildings*, J. Wiley, Chichester (1981).

FORCED SURFACE WAVES IN THE PRESENCE OF FINITE CYLINDRICAL POROUS WALLS

M.A.GORGUI
University Of Alexandria,Faculty Of Science
Mathematics Department,Mohareem Bay
Alexandria,EGYPT

AND

M.S.FALTAS
University Of Bahrain,Mathematics Department
Isa Town,P.O.Box 32038
State Of BAHRAIN

Abstract-A linearised cylindrical wave motion is considered for a fluid of finite depth in the presence of an impermeable cylindrical wall and coaxial porous wall immersed vertically in the fluid. The motion is generated once by the impermeable wall and next by the porous wall. A wave trapping phenomenon is investigated.

1.Introduction

The classical problem of forced two-dimensional wave motion with outgoing wave at infinity generated by a harmonically oscillating vertical wavemaker immersed in water was solved by Havelock [1],Biesel & Suquet [2] and Ursell, Dean & Tu [3]. In these works,the wavemaker was represented by a vertical impermeable plate Chwang [4],Chwang and Li [5],Chwang and Dong [6],Gorgui and Faltas[7] treated wave motion problems in the presence of porous plates.

In the present paper we investigate the effect of porosity on axisymmetric wave motion in fluid of finite depth. The wave trapping phenomenon is discussed.

2.Waves generated by the impermeable wall

We are concerned with the irrotational motion of fluid with free surface which is assumed to be incompressible and inviscid flow under the action of gravity. The motion is induced by an impermeable vertical cylindrical wall of circular cross-section of radius a. The wall assumed to perform radial harmonic oscillations normal to its axis, let its velocity at time t be $U(y)\exp(-i\omega t)$,where $U(y)$ is a complex valued and suitably limited. A coaxial cylindrical porous wall of circular cross-section of radius b (>a) is fixed in the fluid. The resulting motion is therefore axisymmetric and time harmonic with the same angular frequency ω as that of the porous wall.

Let (r,y) be cylindrical polar coordinates with the origin O in the undisturbed free surface such that Oy pointing down into the fluid cinciding with the axis of the porous wall.

We consider the case when the fluid is of finite depth. Let $\phi_j(r,y;t)=\mathrm{Re}[\phi_j(r,y)\exp(-i\omega t)]$, be the velocity potentials where the subscripts j=1,2 refere to the regions $a < r < b$, $r > b$ and the functions ϕ_j satisfy

$$\left[\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial y^2}\right]\phi_j=0 \ , \ y>0 \qquad (2.1)$$

$$K\,\phi_j + \frac{\partial}{\partial y}\phi_j = 0, \text{ on } y=0,\ K=\omega^2/g \qquad (2.2)$$

$$\frac{\partial}{\partial y}\phi_1 = U(y), \text{ on } r=a \qquad (2.3)$$

$$\frac{\partial}{\partial r}\phi_1 = \frac{\partial}{\partial r}\phi_2 \ , \text{ on } r=b \qquad (2.4)$$

We sall assume that the porous wall is made of material with very fine pores. Thus according to Darcy's low [5,8], we have

$$\frac{\partial}{\partial r}\phi_j = iG(\phi_1 - \phi_2), \text{ on } r=b,\ G=\rho\omega b/\mu \ , \qquad (2.5)$$

μ is the dynamic viscosity,ρ is the density and b is a coefficient which has the dimension of length

$$\frac{\partial}{\partial y}\phi_j = 0, \text{ on } y=h, \text{ and} \qquad (2.6)$$

$$\phi_2 \to C\,H_o^{(1)}(\kappa r)\cosh\kappa(h-y) \quad \text{as } r\to\infty \qquad (2.7)$$

Where κ is the real positive root of $k\sinh kh - K\cosh kh = 0$.

3.Solution of the problem

Using the method of separation of variables, solutions of (2.1) satisfying (2.2),(2.6),(2.7) can be written in the form

$$\phi_1=\sum_{n=1}^{\infty}[A_n\,I_o(k_nr)+B_n\,K_o(k_nr)]\cos k_n(h-y)$$
$$+[\alpha\,J_o(\kappa r) + H_o^{(1)}(\kappa r)]\cosh\kappa(h-y) \qquad (3.1)$$

$$\phi_2=\sum_{n=1}^{\infty}C_n\,K_o(k_nr)\cos k_n(h-y) +$$
$$+\dot{\gamma}\,H_o^{(1)}(\kappa r)\cosh\kappa(h-y). \qquad (3.2)$$

From (2.4),(2.5) and (2.3),we get

$$U(y)=\sum_{n=1}^{\infty}\frac{\pi}{2G}k_nC_n\Delta(ik_n)\cos k_n(h-y)$$
$$-\frac{\gamma\kappa}{G}\Delta(\kappa)\cosh\ (h-y) \quad \text{in which}$$

$$\Delta(k)=G\,H_1^{(1)}(ka)+\frac{\pi}{2}k^2b\,H_1^{(1)}(kb)[J_1(kb) - J_1(ka)\times$$
$$H_1^{(1)}(kb)].$$

Since the eigenfunctions $\cosh\kappa(h-y)$ and $\cos k_n(h-y)$ are orthogonal over (0,h) we have

$$C_n = \frac{8Ga_n\cos k_nh}{\delta_n\,\Delta(ik_n)} \ , \quad \gamma = \frac{4\pi Ga_o\cosh\kappa h}{\delta_o\,\Delta(\kappa)}, \text{where}$$

$$\delta_o = 2\kappa h + \sinh 2\kappa h \ , \ \delta_n = 2k_nh + \sin 2k_nh$$

$$a_o = -\frac{1}{\pi\cosh\kappa h}\int_0^h U(y)\cosh\kappa(h-y)\,dy$$

$$a_n = -\frac{1}{\pi\cos k_nh}\int_0^h U(y)\cos k_n(h-y)\,dy$$

Consequently

$$\phi_1=-8i\sum_{n=1}^{\infty}\frac{a_n\cos k_nh}{\delta_n\,\Delta(ik_n)}[k_n^2bK_1(k_nb)I_o(k_nb)-\{iG-k_n^2b\times$$
$$I_1(k_nb)K_1(k_nb)\}K_o(k_nr)]\cos k_n(h-y)-\frac{4\pi a_o\cosh\kappa h}{\delta_o\Delta(\kappa)}\times$$
$$[\tfrac{1}{2}\pi\kappa^2b(H_1^{(1)}(\kappa b))^2J_o(\kappa r)-\{G+\tfrac{1}{2}\pi\kappa^2bH_1^{(1)}(\kappa b)J_1(\kappa b)\}$$
$$\times H_o^{(1)}(\kappa r)\cosh\kappa(h-y), \qquad (3.3)$$

$$\phi_2=-8G\sum_{n=1}^{\infty}\frac{a_n\cos k_nh}{\delta_n\,\Delta(ik_n)}K_o(k_nr)\cos k_n(h-y)$$
$$+\frac{4\pi Ga_o\cosh\kappa h}{\delta_o\Delta(\kappa)}H_o^{(1)}(\kappa r)\cosh\kappa(h-y)\ . \qquad (3.4)$$

The last term on the right hand side of equation (3.4) represents the outgoing wave transmitted through the porous wall.

When the porous wall is completely permeable, the velocity potential in the region $r > a$ is

$$4\pi\sum_{n=1}^{\infty}\frac{a_n\cos k_nh}{\delta_nK_1(k_n)}K_o(k_nr)\cos k_n(h-y)$$
$$+\frac{4\pi a_o\cosh\kappa h}{\delta_o\,H_1^{(1)}(\kappa a)}H_o^{(1)}(\kappa r)\cos\kappa(h-y).$$

For an impermeable wall (G=0),(3.3),(3.4) reduce to

$$\phi_1=4\pi\sum_{n=1}^{\infty}\frac{a_n\cos k_nh}{\delta_n}\,\frac{K_1(k_nb)I_o(k_nr)-I_1(k_nb)K_o(k_nr)}{I_1(k_nb)K_1(k_na)-I_1(k_na)K_1(k_nb)}$$

$$\times\cos k_n(h-y)-4\pi\,\frac{a_o\cosh\kappa h}{\delta_o}\times$$

$$\frac{Y_1(\kappa b)J_o(\kappa r)-J_1(\kappa b)Y_o(\kappa r)}{J_1(\kappa b)Y_1(\kappa a)-J_1(\kappa a)Y_1(\kappa b)}\cosh\kappa(h-y),$$

$\phi_2=0$, as expected. This solution is valid only when the quantity $J_1(\kappa b)Y_1(\kappa a)-J_1(\kappa a)Y_1(\kappa b)$ is diffrent from zero.However,it indicates that when this quantity vanishes,resonance occurs and the linearised theory can not be applied.

4. Waves qenerated by the porous wall

If we let now the porous wall oscillate radially about r=b with velocity $U(y)\exp(-i\omega t)$ while the impermeable wall r=a be kept fixed,then the new problem is the same as stated in section 2 except that (2.3),(2.5) are replaced by

$$\frac{\partial}{\partial r}\phi_1=0,\ \text{on } r=a \tag{4.1}$$

$$\frac{\partial}{\partial r}\phi_j-U(y)=iG(\phi_1-\phi_2) \tag{4.2}$$

In this case we get

$$\phi_1=8i\sum_{n=1}^{\infty}\frac{bk_n^2a_n\cos k_nh\,K_1(k_nb)}{\delta_n\Delta(ik_n)}[I_o(k_nr)K_1(k_na)+$$

$$K_o(k_nr)I_1(k_na)]\cos k_n(h-y)+\frac{2\pi^2\kappa^2ba_o\cosh\kappa h}{\delta_o\Delta(\kappa)}\times$$

$$[J_o(\kappa r)H_1^{(1)}(\kappa a)-H_o^{(1)}(\kappa r)J_1(\kappa a)]H_1^{(1)}(\kappa b)\times$$

$$\cosh\kappa(h-y), \tag{4.3}$$

$$\phi_2=8i\sum_{n=1}^{\infty}\frac{k_n^2ba_n\cos k_nh}{\delta_n}\frac{K_o(k_nr)}{\Delta(ik_n)}[K_1(k_nb)I_1(k_na)-$$

$$I_1(k_nb)K_1(k_na)\cos k_n(h-y)+\frac{2\pi^2\kappa^2ba_o\cosh\kappa h}{\delta_o\Delta(\kappa)}\times$$

$$[J_1(\kappa b)Y_1(\kappa a)-J_1(\kappa a)Y_1(\kappa b)]H_o^{(1)}(\kappa r)\cosh\kappa(h-y) \tag{4.4}$$

When $J_1(\kappa b)Y_1(\kappa a)-J_1(\kappa a)Y_1(\kappa b)=0$,waves are trapped in the bounded region between the two cylinders $a \leq r \leq b$ and no waves radiate away from the wall;liquid simly piles up around the wall.

5. Wave trapping

Let $C\cosh\kappa(h-y)H_o^{(2)}(\kappa r)$ incident normally, proceeding from infinity, the porous wall at r=b and the impermeable wall at r=a are both fixed.The functions ϕ_j are harmonic that satisfy (2.2),(2.6) and

$$\frac{\partial}{\partial r}\phi_1=\frac{\partial}{\partial r}\phi_2=iG(\phi_1-\phi_2),\ \text{on } r=b, \tag{5.1,2}$$

$$\frac{\partial}{\partial r}\phi_1=0,\ \text{on } r=a, \tag{5.3}$$

$$\phi_2\to C\cosh\kappa(h-y)\,H_o^{(2)}(\kappa r)$$
$$+A\cosh\kappa(h-y)\,H_o^{(1)}(\kappa r) \tag{5.4}$$

Consider the functions

$$\psi_j=\phi_j(r,y)-2CJ_o(\kappa r)\cosh\kappa(h-y)$$

These new functions are harmonic satisfying the free surface conditions and

$$\frac{\partial}{\partial r}\psi_1=\frac{\partial}{\partial r}\psi_2 \tag{5.5}$$
$$=iG(\psi_1-\psi_2)+2C\,J_1(\kappa b)\cosh\kappa(h-y), \tag{5.6}$$

$$\frac{\partial}{\partial r}\psi_1=2C\,J_1(\kappa a)\cosh\kappa(h-y), \tag{5.7}$$

and

$$\psi_2\to(A-C)H_o^{(1)}(\kappa r)\cosh\kappa(h-y)\ \text{as } r\to\infty \tag{5.8}$$

Since the present problem is linear,ψ_1,ψ_2 can be obtained by suitable superposition of the results (3.3),(3.4) and (4.3),(4.4) respectively. Hence

$$\phi_1=\frac{2CG}{\Delta(\kappa)}[H_1^{(1)}(\kappa a)J_o(\kappa r)-J_1(\kappa a)H_o^{(1)}(\kappa r)]\times$$
$$\cosh\kappa(h-y)$$

$$\phi_2=-C\,\frac{\Delta'(\kappa)}{\Delta(\kappa)}\,H_o^{(1)}(\kappa r)\cosh\kappa(h-y)+$$
$$C\,H_o^{(2)}(\kappa r)\cosh\kappa(h-y)$$

where $\Delta'(\kappa)=GH_1^{(2)}(\kappa a)+\frac{1}{2}\pi\kappa^2bH_1^{(2)}(\kappa b)[J_1(\kappa b)\times$

$$H_1^{(1)}(\kappa a)-J_1(\kappa a)H_1^{(1)}(\kappa b)]$$

The coeficient of reflection R is defined as the ratio of the amplitude of the reflected wave to the amplitude of the incident wave

$$R=\left|\frac{\Delta'(\kappa)}{\Delta(\kappa)}\right|=\left[\frac{\alpha^2G^2-2\kappa M^2G+\beta^2\kappa^2M^2}{\alpha^2G^2+2\kappa M^2G+\beta^2\kappa^2M^2}\right]^{\frac{1}{2}}<1 \tag{5.9}$$

where

$$\alpha^2=\tfrac{1}{2}\pi\kappa b[J_1^2(\kappa a)+Y_1^2(\kappa a)],$$
$$\beta^2=\tfrac{1}{2}\pi\kappa b[J_1^2(\kappa b)+Y_1^2(\kappa b)]$$
$$M=\tfrac{1}{2}\pi\kappa b[J_1(\kappa b)Y_1(\kappa a)-J_1(\kappa a)Y_1(\kappa b)]$$

For an impermeable wall,the incident wave is totally reflected by it.We get the same situation when the wall is completly permeable but now the wave is totally reflected at the impermeable wall at r=a. We note also that when M=0 i.e.when a and b satisfy the eqation

$$J_1(\kappa b)Y_1(\kappa a)-J_1(\kappa a)Y_1(\kappa b)=0\ ,$$

the incident wave is totally reflected irrespective of the vale of G .By simple differentiation of (5.9), we note that the value of R for any fixed a and b reduces to a minimum

$$R_{min}=\left[\frac{\alpha\beta-M}{\alpha\beta+M}\right]^{\frac{1}{2}},$$

when $\frac{G}{\kappa}=\frac{M\beta}{\alpha}$, this minimum value vanishes when $\alpha\beta=M$ or when a and b satisfy the equation

$$J_1(\kappa a)J_1(\kappa b)+Y_1(\kappa a)Y_1(\kappa b)=0 \tag{5.10}$$

in this case $\frac{G}{\alpha}=\beta^2$ Under these circumstances the porous wall acts as an efficient wave absorber or eleminator for the incident waves, i,e. for the values of $G/\kappa=\beta^2$ and where a and b satisfy equation (5.10), there is a wave trapping phenomenon that is,wave will be trapped inside the region $a \leq r \leq b$.

References

[1] Havelock, T.H. Phil.Mag.8(1929),569-576
[2] Biesel,F.& Suquet,F. Houille Blanche 6(1951),147-165,475-496,723-737.
[3] Ursell,F.,Dean,R,G.& Yu,Y.S. J.Fluid Mech. 7(1960),33-52
[4] Chwang,A.T.J.Fluid Mech.132(1983),395-406
[5] Chwang,A.T.& Li,W. J.Eng.Math.17(1983), 301-313
[6] Chwang,A.T.& Dong,Z. F oc.15th.Symposium on Naval hydrodynamics.pp407-417.Washington:National Academy press 1985
[7] Gorgui,M.A.& Faltas,M.S.Acta Mechanica 79(1989),259-275
[8] Talyor,G.I.Proc.R.Soc.Lond.A234(1956),456-475

STEADY PLANE ORTHOGONAL MHD FLOWS

O.P. Chandna and P.V. Nguyen
Department of Mathematics and Statistics
University of Windsor, Windsor, Ontario
Canada N9B 3P4

Abstract - Steady plane magnetohydrodynamic flow of a viscous incompressible fluid of infinite electcical conductivity is governed by

$$\begin{aligned} \operatorname{div} \underset{\sim}{V} &= 0 \\ \rho(\underset{\sim}{V}.\operatorname{grad})\underset{\sim}{V} + \operatorname{grad} p &= \mu \nabla^2 \underset{\sim}{V} + \mu^*(\operatorname{curl} \underset{\sim}{H}) \times \underset{\sim}{H} \\ \operatorname{curl}(\underset{\sim}{V} \times \underset{\sim}{H}) &= \underset{\sim}{0} \\ \operatorname{div} \underset{\sim}{H} &= 0 \end{aligned}$$

where $\underset{\sim}{V} = (u, V)$ denotes the velocity vector, $\underset{\sim}{H} = (H_1, H_2)$ the magnetic field vector, p the pressure, ρ the constant fluid density, μ the constant coefficient of viscosity and μ^* the constant magnetic permeability. The magnetic field vector $\underset{\sim}{H}$ is given by the solution of

$$\begin{aligned} \underset{\sim}{V} \cdot \underset{\sim}{H} &= 0 , \\ \underset{\sim}{V} \times \underset{\sim}{H} &= A \underset{\sim}{K} \end{aligned}$$

when the magnetic field is orthogonal to the velocity field. Writing the governing equations in x, y coordinates and recasting these in new independent variables $z = x + iy$ and $\bar{z} = x - iy$, we find that:

'If $\psi(z, \bar{z})$ is the streamfunction of the flow, then $\psi(z, z)$ must satisfy

$$\psi_{z\bar{z}z\bar{z}} - \frac{1}{\upsilon}\mathrm{Im}\{\psi_{z\bar{z}z}\psi_z\} - \frac{\mu^* A^2}{32\mu}\{\mathrm{Re}\{[\mathrm{Im}\{(\frac{1}{\psi_z^2\psi_{\bar{z}}^2})_z \psi_{\bar{z}}\}]_z \psi_{\bar{z}}\}$$

$$+ \psi_{z\bar{z}}\mathrm{Im}\{(\frac{1}{\psi_z^2\psi_{\bar{z}}^2})_z \psi_{\bar{z}}\}\} = 0$$

$$\frac{\psi_{zz}}{\psi_z\psi_{\bar{z}}} + \mathrm{Re}\{(\frac{1}{\psi_z\psi_{\bar{z}}})_z \psi_{\bar{z}}\} = 0 \text{ '}$$

We have studied exact integrals for four different flow geometries and Hamel's problem [1] in this work. The novelty of this work is in its approach since the Hamel's problem that has been investigated here was also investigated by Chandna and Toews [2]. The approaches used in the previously published works required the transformation of the flow equations to curvilinear coordinates when the streamlines and their orthogonal trajectories formed the coordinate net. No such transformation is required in the present complex variable approach.

The streamlines and their orthogonal trajectories form an isometric net for incompressible and irrotational steady plane flows. G. Hamel investigated those steady plane rotational fluid motions for which the streamlines and their orthogonal trajectories form an isometric net. This problem is called Hamel's problem. Martin [3] gave new formulation of the Navier-Stokes equations and studied Hamel's problem as an application of his approach. His method also required the transformation of the flow equations to the streamline curvilinear coordinates.

Complex variable technique employed in this paper is well known for the analysis of fluid dynamic problem. Wan-Lee Yin [4], Ratip Berker [5] and Stallybrass [6] have employed this technique in their recent researches.

References

[1] G. Hamel: Jber. Deutsch. Math. Verein 25 (1916)34.
[2] O.P. Chandna and H. Toews: Q. Appl. Math 34 (1977) 331.
[3] M.H. Martin: Arch. Ration. Mech. & Anal. 41 (1971) 266.
[4] Wan-Lee-Yin: Q. Appl. Math. 42 (1984) 31.
[5] Ratip Berker: Comptes Rendus de L'Academie des Sciences de Paris, 242 (1956) 342.
[6] M.P. Stallybrass: Lett. Appl. Engng. Sci. 21 (1983) 179.

ON THE ANALYSIS OF SUPERHARMONIC OSCILLATIONS

J. J. Wu
US Army Research Office
Research Triangle Park, NC 27709 USA

ABSTRACT - This paper presents an analysis for the superharmonics of a forced nonlinear vibration problem involving small parameters, using a generalized harmonic balance method. A nonlinear ordinary differential equation with several nonlinear terms and a periodic forcing function is considered. For the case of superharmonic oscillations of order 2, the key equtions for the obtaining the information on the superharmonics will be derived, including a new, nonlinear ordinary differential equation of a slow varying function compared with the original dependent variable. Using these equations, the steady state solution and its stability behavior can be calculated. Results for a special set of parameters are obtained, including a stable node for the steady state solution and the associated van del Pol plane.

1. INTRODUCTION

It is well known that nonlinearities can cause sub- and super-harmonic excitations in vibratory systems. The analytical understanding of such phenomena is often difficult to obtain. It has been shown that the method of multiple scales can be used to solve such problems as demonstrated in several papers by Nayheh [1,2]. However, the procedures involved are quite complicated and requires recursive solution of differential equations, the elimination of secular terms and reconstitution, all of which are nontrivial procedures. More recently, in a paper by Noble and Hussain [3], an expansion method was introduced together with suggestions of several other approaches which may be used as alternatives to obtain pertinent information. One of these is the genralized harmonic balance method (GHB) [4,5,6]. This variant of the harmonic balance method consists of two parts: first, to derive the form of solution using only the basic steps of multiple scales, and then, solve for the coefficients of various harmonics. In this approach, the elimination of the secular terms is accomplished implicitly, thus avoiding the trouble of solving recursive differential equations. This paper begins with a general nonlinear ordinary differential equation with several nonlinear terms and a periodic forcing function, a specific case of superharmonic oscillations of order 2 will be investigated. Next, the key equations are derived, from them the essential information on the superharmonics can be obtained. Finally Numerical results are presented on the steady solution and the stability behavior for a special sets of parameters.

2. DERIVATION OF THE KEY EQUATIONS

We shall consider the following rather general differential equation:

$$d^2u/dt^2+u+2\varepsilon\mu(du/dt)+\varepsilon\alpha_2u^2+\varepsilon^2\alpha_3u^3 \\ +\varepsilon\alpha_4(du/dt)^2+\varepsilon^2\alpha_5u(du/dt)^2=2f\cos(\Omega t) \quad (1)$$

where u(t) is the unknown function μ and α_k, k=2,3,4,5 and 6, are given constants, ε is the small perturbation parameter; f and Ω pertain to the magnitude and frequency of the forcing function. For superharmonics of order 2, one has

$$2\Omega = \omega = \omega_0 + \varepsilon\sigma = 1 + \varepsilon\ \sigma \quad (2)$$

where ω is the "fundamental" frequency of the nonlinear vibration, which is a perturbation from that of the linearalized system ω_0, taking to be unity in (2) without a loss of generality.

We shall derive a two-term approximate solution $u=u_0+\varepsilon u_1$ for equation (1). Using a procedure described previously in [4,5], it can be shown easily that that the final form of the solution u, which is good tc the order of ε must have the following form:

$$u=\varepsilon U_0+[(U_1A+U_2A^2)+\varepsilon(U_3A^3+U_4A^4)+cc \quad (3)$$

where cc stands for the complex conjugate. The following symbals has been introduced:

$$A=\exp(it/2), \qquad S=\exp(i\varepsilon\sigma t/2) \quad (4)$$

Eq. (1) can then be written as

$$d^2u/dt^2+u+2\varepsilon\mu(du/dt)+\varepsilon\alpha_2u^2+\varepsilon^2\alpha_3u^3 \\ +\varepsilon\alpha_4(du/dt)^2+\varepsilon^2\alpha_5u(du/dt)^2=fSA^2+cc \quad (1')$$

Here we note that S is a slow varying function compared with A in the sense that while dA/dt is of O(1), dS/dt is of O(ε). We shall also use the fact that

$$\overline{A} = e^{-it/2}, \text{ and } A\overline{A} = 1 \quad (5)$$

where an overbar denotes the complex conjugate. The procedure here is to substitute (3) in (1′) and set to zero the coefficients of A^k, k=0,1 and 2, since any higher harmonics will be of O(ε^2) or higher according (4). We first obtain the following approximate expressions (in other words, the right hand side should have added "+ terms of O(ε^3) and higher" in each of these equations):

$$du/dt=(dU_1/dt+iU_1)A \\ +\varepsilon[dU_0/dt+(dU_2/dt+2iU_2)A^2] + cc \quad (6)$$

$$d^2u/dt^2=\varepsilon d^2U_0/dt^2+(d^2U_1/dt^2+2idU_1/dt-U_1)A \\ +\varepsilon(d^2U_2/dt^2+4idU_2/dt-4U_2)A^2 + cc \quad (7)$$

$$u^2=2U_1\overline{U}_1+U_1{}^2A^2+2\varepsilon(\overline{U}_1U_2+U_0U_1)A +cc \quad (8)$$

Since u^3 appears with a coefficient of ε^2 in (1), one only needs to keep terms of O(1) in the expansion:

$$u^3=3U_1{}^2\overline{U}_1A + cc \quad (9)$$

Similarly, one keeps O(ε) teims in $(du/dt)^2$, but only O(1) terms in $u(du/dt)^2$:

$$(du/dt)^2= 2U_1\overline{U}_1-(U_1{}^2A^2+cc) \quad (10)$$

$$u(du/dt)^2=U_1{}^2U_1A + cc \quad (11)$$

We now substitute (4) and (6)-(11) in (1'), collect terms of like power of A^k, k=0,1 and 2, and then set the coefficients to zero. The resulting equations, for the coefficients of A^0, A^1 and A^2 respectively, are:

$$\varepsilon[U_0+2(\alpha_2+\alpha_4)U_1\overline{U}_1+(1/2)(4\alpha_2+\alpha_4)U_2\overline{U}_2]=0 \tag{12}$$

$$3U_1/4-fS+idU_1/dt+i\varepsilon\mu U_1+\varepsilon(2\alpha_2+\alpha_4)U_1\overline{U}_2=0 \tag{13}$$

$$\begin{aligned}&2i(dU_2/dt+\varepsilon\mu U_2)+\varepsilon(4\alpha_2-\alpha_4)U_1{}^2/4+d^2U_2/dt^2\\&+\varepsilon(2\mu dU_2/dt+i\alpha_4 U_1 dU_1/dt\\&+\varepsilon^2[2\alpha_2U_0U_2+(2\alpha_2+3\alpha_4/2)\overline{U}_1U_3+2(\alpha_2+2\alpha_4)\overline{U}_2U_4\\&+(3\alpha_3+\alpha_5)U_2{}^2\overline{U}_2+(6\alpha_3+\alpha_5/2)U_1\overline{U}_1U_2)]=0\end{aligned} \tag{14}$$

$$-5\varepsilon U_3/4+\varepsilon(2\alpha_2-\alpha_4)U_1U_2 = 0 \tag{15}$$

$$-3\varepsilon U_4+\varepsilon(\alpha_2-\alpha_4)U_2{}^2 = 0 \tag{16}$$

From (12), (15) and (16), U_0, U_3 and U_4 can be solved directly in terms of U_1 and U_2:

$$U_0 = -2(\alpha_2+\alpha_4)U_1\overline{U}_1-(1/2)(4\alpha_2+\alpha_4)U_2\overline{U}_2 \tag{17}$$

$$U_3=(4/5)(2\alpha_2-\alpha_4)U_1U2 \tag{18}$$

$$U_4=(\alpha_2-\alpha_4)U_2{}^2/3 \tag{19}$$

In equation (13) and (14), however, it is observed that some terms are of one order of ε greater than the others. The terms of higher order in ε can thus be less accurate than others and still yield the same order of approximation in these equations. One then can solve these equation first using only the dominant terms. Then, substitute the results back into the terms of higher order in ε, solve the full equations and obtain improved results. The immediate purpose here is to reduce (16) into a first order differential equation in U_2 and express all the other U_ks in terms of U_2.

Using the dominant terms in (13) and (14), one has

$$U_1 = 4fS/3 \tag{20}$$

$$2i(dU_2/dt+\varepsilon\mu U_2)+\varepsilon(4\alpha_2-\alpha_4)U_1{}^2/4 = 0 \tag{21}$$

Equation (20) is used in the terms of order ε in (13) to yield the improved U_1:

$$\begin{aligned}&U_1=4fS/3+(1/9)\varepsilon[8(\sigma-2i\mu)fS\\&-16(2\alpha_2+\alpha_4)fS\overline{U}_2 = 0\end{aligned} \tag{22}$$

Now, the terms in (16), which are of higher order in ε, contain such quantities as d^2U_2/dt^2, dU_2/dt, dU_1/dt, U_1, U_0, U_3, U_4. These expressions can be obtained by using (21), (22), their differentiations (for $d^2U_{.}/dt^2$ and dU_1/dt), (17), (18) and (19). The final form of (16) can be written as th^ following:

$$\begin{aligned}&2idU_2/dt+\varepsilon(2i\mu U_2+c_1f^2S^2U_2)\\&+\varepsilon^2[c_2U_2{}^2U_2+c_{34}f^2S^2+(c_5f^2-\mu^2)U_2)] = 0\end{aligned} \tag{23}$$

where

$$c_1 = 4(4\alpha_2-\alpha_4)/9$$

$$c_2 = (9\alpha_3+3\alpha_5-10\alpha_2{}^2-10\alpha_2\alpha_4-4\alpha_4{}^2)/3$$

$$c_{34} = c_3 + ic_4$$

with

$$c_3 = 2\sigma(20\alpha2-17\alpha_4)/27 \tag{24}$$

$$c_4 = -2\mu(52\alpha_2-13\alpha_4)/27$$

and

$$\begin{aligned}c_5 = &(1440\alpha_3+120\alpha_5-1472\alpha_2{}^2\\&-368\alpha_2\alpha_4-128\alpha_4{}^2)/135\end{aligned}$$

The key equations (4), (16), (17), (18), (19) and (22) can be further simplified by the following change of variables. Let

$$U_k = V_kS^k,\quad V_k = U_kS^{-k},\quad k=0,1,..4 \tag{25}$$

where S has been defined in (6). One also has

$$dU_k/dt = dV_k/dt + ik\varepsilon\sigma V_k/2 \tag{26}$$

In terms of V_k, equations (4), (16), (17), (18), (19) and (22) become respectively

$$u = \varepsilon V_0+[V_1B+V_2B^2+\varepsilon(V_3B^3+V_4B^4)+c\bar{c}] \tag{27}$$

with

$$V_0 = -(32/9)(\alpha_2+\alpha_4)f^2-2(\alpha_2+\alpha_4)V_2\overline{V}_2 \tag{28}$$

$$V_1=4f/3+(1/9)\varepsilon[8(\sigma-2i\mu)f-16(2\alpha_2+\alpha_4)fV_2 = 0 \tag{29}$$

$$V_3=(4/5)(2\alpha_2-\alpha_4)V_1V_2 \tag{30}$$

$$V_4=(\alpha_2-\alpha_4)V_2{}^2/3 \tag{31}$$

and,

$$\begin{aligned}&2idV_2/dt+\varepsilon(-2\sigma+2i\mu+c_1f^2)V_2\\&+\varepsilon^2[c_2V_2{}^2\overline{V}_2+c_{34}f^2+(c_5f^2-\mu^2)V_2)] = 0\end{aligned} \tag{32}$$

where, in (29),

$$B = SA = \exp[(1+\varepsilon\sigma/2)t] = e^{i\Omega t} \tag{33}$$

Hence the original differential equation (1) has been reduced to (32), where V_2 is the unknown function. Once V_2 is solved, other V_ks can be obtained from (28) through (31). Then u(t) is given by (27).

To illustrate what kind of information one can extract from the equations derived so far, we shall obtain the magnitude for a superharmonic in the steady state solution and determine the stability of such a solution. First, we shall write the needed equations in terms of real variables. To this end, let

$$V_2=V_{2R}+iV_{2I}=\rho_2\ \exp(i\gamma_2)$$

and

$$V_2=(x-iy)/2 \tag{34}$$

where now ρ_2, γ_2 V_{2R}=x/2 and V_{2I}=-y/2 are all real functions of t. One also has

$$dV_2/dt=(dx/dt-idy/dt)/2 \tag{35}$$

Note that we have introduced two new variables x and y such that

$$x=2V_{2R},\quad y=-2V_{2I} \tag{36}$$

to save some writing. Substitute (34) and (35) in (32) and separate the real and imaginary part, one has two equations for two real variables x and y:

$$dx/dt+\varepsilon[\mu x+\sigma y]+\varepsilon^2[c_4 f^2$$

$$-c_2(x^2+y^2)y/8+(c_5 f^2-\mu^2)y/2] = 0 \qquad (37a)$$

$$dy/dt+\varepsilon[\mu y-\sigma x+c_1 f^2]+\varepsilon^2[c_3 f^2$$

$$+c_2(x^2+y^2)x/8+(c_5 f^2-\mu^2)x/2] = 0 \qquad (37b)$$

For steady state solutions, we require that the amplitudes and phase angles of various harmonic components to be constant with respect to time t,

$$d\rho k/dt = 0,\ d\gamma k/dt = 0,\ k=0,1..,4 \qquad (38)$$

In particular,

$$d\rho_2/dt = 0,\quad d\gamma_2/dt = 0 \qquad (39a)$$

and, what is equivalent:

$$dx/dt=0,\quad dy/dt=0 \qquad (39b)$$

It should be noted that (39a) actually also quarantee the validity of (38) for k other than 2. This fact can be easily observed from the relations of (28)-(31), which relate V_k, k=0,1,3 and 4, to V_2.

Now, substitute (39b) in (37), one has

$$\mu x+\sigma y+\varepsilon[c_4 f^2-c_2(x^2+y^2)y/8$$

$$+(c_5 f^2-\mu^2)y/2] = 0 \qquad (40a)$$

$$\mu y-\sigma x+c_1 f^2+\varepsilon[c_3 f^2+c_2(x^2+y^2)x/8$$

$$+(c_5 f^2-\mu^2)x/2] = 0 \qquad (40b)$$

Some numerical results will be presented in determining the presence of superharmonic oscillations for the following given set of parameters:

$$\alpha_2=0.3,\ \alpha_3=0.1,\ \alpha_4=0.,\ \alpha_5=0.,$$

$$\mu=2.0,\ \sigma=3.0,\ f=2.0 \qquad (41)$$

This is a very simple case due to the fact that c_2 vanishes as can be seen from (24). Thus (40) become linear and the solution can be easily obtained as

$$x=0.1824,\ y=-0.0418 \qquad (42)$$

Hence, from (33), the magnitude of the superharmonic oscillation of order 2, ρ_2 is

$$\rho_2=0.5(x^2+y^2)=0.3754 \qquad (43)$$

Next, equations (37) are integrated numerically. The result is the so called van del Pol plane [7] as show in Figure 1. As indicated in this plot, solutions converge to the steady state solution obtained above as the time increases. Hence the steady state solution is stable and the point "A" of (42) is known as a stable node.

Results for more general cases will be reported in the future.

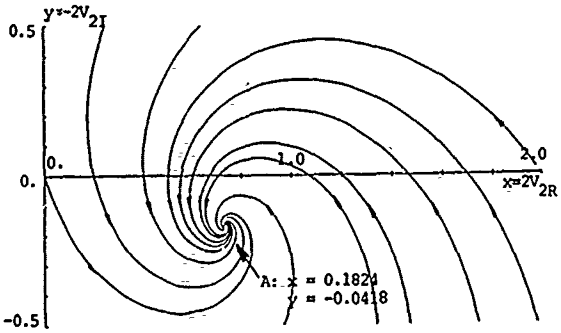


FIGURE 1. The van del Pol plane for the superharmonics of order 2 for the set of parameters given in Eqn.(41). Point "A" shown is a stable node.

REFERENCES

[1] A. H. Nayfeh, The response of single degree of freedom systems with quadratic and cubic non-linearities to a subharmonic excitation, Journal of Sound and Vibration (1983), Vol. 89(4), pp.457-470.

[2] A. H. Nayfeh, Perturbation Methods in Nonlinear Dynamics, Lecture Notes in Physics: Nonlinear Dynamics Aspects of Particle Accelerators - Proceedings of the Joint US-CERN School on Particle Accelerators, Editors: J. M. Jowett, M. Month and S. Turner, Spring-Verlag, 1985, pp.238-314.

[3] B. Noble and M. A. Hussain, Multiple Scaling and a Related Expansion Method, with Applications, Lasers, Molecules and Methods (J. O. Hirschfelder, R. E. Wyatt and R. D. Coalson, Eds.), John Wiley & Sons,1989, pp.83-136.

[4] M. A. Hussain, B. Noble and J. J. Wu, Using Macsyma in a Generalized Harmonic Balance Method for a Problem od Forced Nonlinear Oscillation, Proc. Sixth Army Conference on Applied Mathematics and Computing (held 31 May - 3 June 1988, Univ. of Colarado, Boulder, Colorado), 1989, pp.713-732.

[5] B. Noble, M. A. Hussain and J. J. Wu, A Generalized Harmonic Balance Method for a Forced Nonlinear Oscillation - Numerical Solution Formulation and Results, Proc. Seventh Army Conference on Applied Mathematics and Computing (held 6-9 June 1989, U.S. Military Academy, West Point, New York), 1990, pp.837-861.

[6] J. J. Wu, On the Analysis of Subharmonic Oscillations, Submitted for publication.

[7] D. W. Jordan and P. Smith, Nonlinear Differential Equations, Second Edition, Oxford University Press, 1986, p.183.

DIFFRACTION ON A PERIODIC SURFACE

Andrew G.Mikheev
Institute for Problems in Mechanics
USSR Academy of Sciences
Pr. Vernadskogo 101, Moscow 117526, USSR

and

Aleksey S.Shamaev
Institute for Problems in Mechanics
USSR Academy of Sciences
Pr. Vernadskogo 101, Moscow 117526, USSR

Abstract In this paper the two-dimensional problem of diffraction of a plane electromagnetic wave on a smooth 2π-periodic surface is considered. The numerical method solving the problem of diffraction is developed.

1. Mathematical formulation of the problem.

The unknown function u satisfies the Helmholtz equation

$$\Delta u + k^2 \cdot u = 0 \qquad (1)$$

in the region $\Omega = \{ (x,y) \mid -\infty < x < f(y),\ 0 \le y \le 2\pi \}$.
Here k is wavenumber, $k=\frac{\omega}{c}$, f(y) is smooth 2π-periodic function.

The boundary condition for the .function u is :

$$\frac{\partial u}{\partial n} - h \cdot u(f(y),y) = 0 \qquad (2)$$

In the region $x < x_0 = \inf\limits_{[0,2\pi]} f(y)$ the radiation condition

$$u = e^{ik(x\cdot\cos\alpha + y\cdot\sin\alpha)} +$$

$$+ \sum_{n=-\infty}^{+\infty} T_n \cdot e^{-i\gamma_n x} \cdot e^{i\lambda_n y} \qquad (3)$$

is imposed on u . Here α is the angle between the wave vector of incident wave and x - axis, $\lambda_n = k\cdot\sin\alpha + n$, $\gamma_n = \sqrt{k^2 - \lambda_n^2}$, Re $\gamma_n \ge 0$ Im $\gamma_n \ge 0$. T_n are unknown amplitudes of scattered plane waves.

The function u is also assumed to satisfy the Flocke conditions:

$$u(x,\ 2\pi) = u(x,\ 0)\cdot e^{it} \qquad (4)$$

$$\frac{\partial u}{\partial y}(x,\ 2\pi) = \frac{\partial u}{\partial y}(x,\ 0)\cdot e^{it} \qquad (5)$$

where $t = 2\pi k\cdot\sin\alpha$.

2. Numerical algorithm solving the diffraction problem.

With the help of Green's function of Flocke canal

$$G(M,P) = \frac{i}{2} \cdot \sum_{m=-\infty}^{+\infty} \frac{e^{i\lambda_m \delta y} \cdot e^{i\gamma_m |\delta x|}}{\gamma_m} \qquad (6)$$

(here $\delta x = x_M - x_P$, $\delta y = y_M - y_P$)
the problem is reduced to the one-dimensional integral equation for the u(f(y),y) :

$$\frac{1}{2}\cdot u(f(y_M),y_M) - \frac{1}{2\pi}\cdot\int_0^{2\pi} K\left[y_M,y_P\right]\cdot I(y_P)\cdot$$

$$\cdot u(f(y_P),y_P)dy_P = e^{ik(f(y_M)\cdot\cos\alpha + y_M\cdot\sin\alpha)} \qquad (7)$$

Here $I(y) = \sqrt{1 + [f'(y)]^2}$,

$$K\left[y_M,y_P\right] = h\cdot G(f(y_M),y_M,f(y_P),y_P) -$$

$$- \frac{\partial G}{\partial n}(f(y_M),y_M,f(y_P),y_P)$$

The integral equation (7) was solved with the help of method-of-moments :

Following [1], let us divide the segment $[0,2\pi]$ into N equal length segments, using points y_i ($y_0=0$, $y_N=2\pi$). Consider functions :

$$\phi_i(y) = \begin{cases} 1, & y\in[y_{i-1},y_i] \\ 0, & y\notin[y_{i-1},y_i] \end{cases}$$

Let us seek an approximate solution of equation (7) in the following form :

$$\Psi^N(y) = \sum_{i=1}^{N} D_i^N \cdot \phi_i(y) \qquad (8)$$

where coefficients D_i^N are to be determined.

Function $\Psi^N(y)$ is assumed to satisfy equation (7) in points $y_{i-\frac{1}{2}} = \frac{1}{2}\cdot(y_{i-1} + y_i)$.

This gives algebraic equations for D_i^N coefficients determination.

The expressions (9)-(14) give the well convergent series, which gives us the method for calculating the kernel of the integral equation (7).

$$G\ (\ M,P\ ) = \frac{i\cdot e^{i\lambda_o \delta y}\cdot e^{i\gamma_o|\delta x|}}{2\gamma_o} +$$

$$+ \sum_{n=1}^{\infty} \Big[e^{i\frac{t}{b}\delta y} \cdot \Big(ch(\tfrac{t}{b}|\delta x|)\cdot\Phi_1(n,\delta x,\delta y) -$$

$$- i\cdot sh(\tfrac{t}{b}|\delta x|)\cdot\Phi_2(n,\delta x,\delta y) \Big) +$$

$$R(n,\delta x,\delta y) \Big] \qquad (9)$$

Here $M = (x_M,y_M)$, $P = (x_P,y_P)$,

$b=2\pi$,

$$\Phi_1(n,\delta x,\delta y) = \frac{\cos(n\delta y)}{n}\cdot e^{-n|\delta x|} \qquad (10)$$

$$\Phi_2(n,\delta x,\delta y) = \frac{\sin(n\delta y)}{n}\cdot e^{-n|\delta x|} \qquad (11)$$

Expressions (10), (11) can be summed:

$$\sum_{n=1}^{\infty} \Phi_1(n,\delta x,\delta y) = -\ln 2 + \frac{|\delta x|}{2} -$$

$$- \frac{1}{2}\cdot \ln\left[\mathrm{sh}^2\left(\frac{\delta x}{2} \right) + \sin^2\left(\frac{\delta y}{2} \right) \right] \qquad (12)$$

$$\sum_{n=1}^{\infty} \Phi_2(n,\delta x,\delta y) =$$

$$= \mathrm{arctg} \left[\frac{\sin(\delta y)}{e^{|\delta x|} - \cos(\delta y)} \right] \qquad (13)$$

$$\frac{\partial G}{\partial n_p} = \frac{\partial}{\partial n_p} \frac{i\cdot e^{i\lambda_0 \delta y}\cdot e^{i\gamma_0 |\delta x|}}{2\gamma_0} +$$

$$\sum_{m=1}^{\infty} \Bigl\{ e^{i\frac{t}{b}\delta y} \cdot \Bigl[\mathrm{ch}(\tfrac{t}{b}|\delta x|)\cdot \frac{\partial \Phi_1(m)}{\partial n_p} -$$

$$i\cdot \mathrm{sh}(\tfrac{t}{b}|\delta x|)\cdot \frac{\partial \Phi_2(m)}{\partial n_p} - k^2\cdot \frac{|\delta x|}{2}\cdot \Bigl(-\Phi_1(m) \cdot$$

$$(\mathrm{sign}(\delta x)\cdot \mathrm{ch}(\tfrac{t}{b}|\delta x|)\cdot n_x ++ i\cdot \mathrm{sh}(\tfrac{t}{b}|\delta x|)\cdot n_y) +$$

$$+\Phi_2(m)\cdot (-\mathrm{ch}(\tfrac{t}{b}|\delta x|)\cdot n_y + i\cdot \mathrm{sh}(\tfrac{t}{b}|\delta x|)\cdot \mathrm{sign}(\delta x)\cdot$$

$$\cdot n_x) \Bigr) \Bigr] + Q(m,M,P) \Bigr\} \qquad (14)$$

R, Q satisfy the expressions (15)

$$|\, R(n,\delta x,\delta y)\, | < \frac{C}{n^2};\ |\, Q(m,M,P)\, | < \frac{C}{m^2} \qquad (15)$$

The kernel of the integral equation contains logarithmic singularity, which is expressed in explicit form in (12).

3. Numerical examples.

Figures 1, 2 show a peculiar example of distribution of electromagnetic surface current (function u(f(y),y)).

Here $k = \frac{1}{2}$, $\alpha = 75^\circ$, $h = 0$

$f(y) = -\frac{1}{2}\cdot\sin y$.

Parameter N = 30 (see (8))

Energy error equals $4\cdot 10^{-4}$.

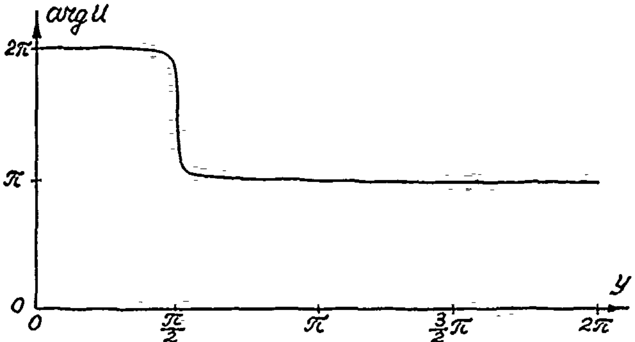


Fig. 1. Phase of surface current.

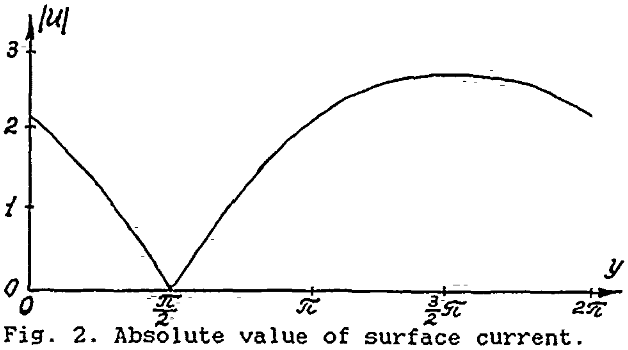


Fig. 2. Absolute value of surface current.

Figures 3, 4 show a typical example of distribution of electromagnetic surface current.

Here $k = \frac{3}{2}$, $\alpha = 60^\circ$, $h = (1+i)\cdot 0.075$,

$f(y) = -\frac{1}{2}\cdot\sin y$.

Parameter N = 30. Energy error equals $2\cdot 10^{-2}$. Share of reflecting energy = 0.28 .

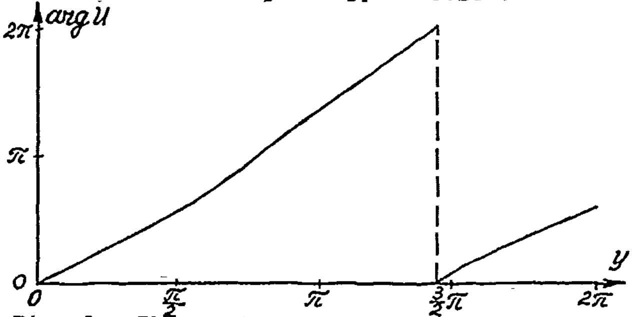


Fig. 3. Phase of surface current.

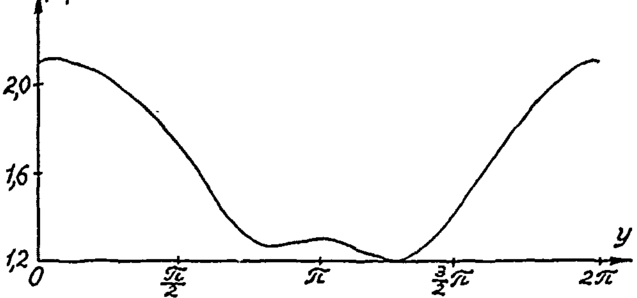


Fig. 4. Absolute value of surface current.

REFERENCES

1. GALISHNICOVA T. N., IL'INSKIY A. S. Numerical methods in diffraction problems. Mosk. Univ. 1987 (In Russian).

Simulation of Distributed Feedback Dye Laser by Computer

by J. Seres

Department of Physics, Juhász Gyula College
Boldogasszony sgt. 6. PO Box 396 Szeged, H-6701 Hungary

Abstract-. A numerical method has been developed for computing coupled partial differential equations describing distributed feedback dye lasers. This method is founded on the Euler method, but is faster about 30 times. The necessary compatition on the time and space axes has been determined for the required accuracy computations. Calculations have been made by the novel method and the obtained results have been compared with earlier publised measurements. The computed results have shown good agreements with measured values.

I. INTRODUCTION

The arrangements and behaviours of DFDLs have been extensively studied since 1971. A typical arrangement of the DFDL and several tuning possibilities are shown by [3]. The processes happening in DFDLs may be described by coupled differential equations (rate equations) [1,3,6-9]. For the most part these equations contain physical quantities depending on only the time variable. Many calculations were made and their results were compared with measured values in last years. The experiences show if the pulse duration of the exciting laser and DFDL is longer many times than the time calculated by the laser length and the refractive index then computed values approximate the values of measurements [1,2,3]. If these conditions don't exist the measured values are fully different from calculated ones. In this case the calculations give exact values if the physical quantities describing DFDL depend on space variable too [9]. For two variables the running time is many times longer than it for one variable. A novel algorithm has been developed to reduce the running time. This algorithm based on the Euler method, but faster than it many times.

II. MODEL OF DFDL

Coupled partial differential equations are shown below (1,2). These equations noted down with the help of publised equations in [1,3,5,8,9] describe the processes in DFDLs.

$$\frac{\partial N(x,t)}{\partial t} = I_p(t)\cdot\sigma_p\cdot\left[N_o - N(x,t)\right] - \frac{N(x,t)}{\tau} - $$

$$- \sigma_o\cdot N(x,t)\cdot\left[I^+(x,t) + I^-(x,t)\right] \; ; \qquad (1)$$

$$\pm\frac{\partial I^\pm(x,t)}{\partial x} + \frac{n}{c}\cdot\frac{\partial I^\pm(x,t)}{\partial t} = \frac{\Omega\cdot N(x,t)}{\tau} + (\sigma_o-\sigma_a)\cdot$$

$$\cdot N(x,t)\cdot\left[I^\pm(x,t) + \frac{V^2}{4}\cdot\left[I^\mp(x,t) - I^\pm(x,t)\right]\right] \quad (2)$$

The meaning of the symbols are as follows:

N_0 :the density of dye molesules $[2.1\cdot10^{24}m^{-3}]$,

$N(x,t)$:the density of molecules in the S_1 excited state $[\text{molecules}\cdot m^{-3}]$,

$I^\pm(x,t)$:the density of the DFDL photon current propageting into the +x and -x direction, respectively $[\text{photons}\cdot m^{-2}s^{-1}]$,

$I_p(t)$:the density of the pump photon current $[\text{photons}\cdot m^{-2}s^{-1}]$,

σ_p :the absorption cross section from S_0 at the pumping wavelength $[2.4\cdot10^{-24}m^2]$,

σ_e :the stimulated emission cross section from S_1 to S_0 state at the lasing wavelength $[1.4\cdot10^{-20}m^2]$,

σ_a :the excited state absorption cross section from S_1 to S_2 at the lasing wavelength $[0.7\cdot10^{-20}m^2]$,

τ :the fluorescence lifetime of the S_1 state [4 ns],

n :the refractive index of the dye solution [1.44],

c :the speed of light in vacuum,

V :the visibility of the amlitude-phase grating in the excited volume [1],

Ω :the factor determining that fraction of the spontaneous emission which propagates into the angular and spectral range of the DFDL beam.

The value of Ω is calculated as $\Omega = \dfrac{b\cdot a}{\pi\cdot L^2\cdot S}$, where L is the length of DFDL, $a = (N_0\cdot\sigma_p)^{-1}$ is the penetration depth of the pumping beam into the dye solution, b is the height of the excited volume, and S is the spectral factor determining that fraction of the spontaneus emission, which falls into the DFDL bandwidth [1,3]. This parameters described the dye laser what contained the Rhodamine 6G dissolved in methanol and what was excited by N_2 laser. The wavelength of N_2 laser was 337.1 nm, and the pulse duration was 3.5 ns.

III. LIMIT OF Δx AND Δt

Calculations have been made to determine the limits of the distributing intervals on both time and space axes. The rate equations depending only time variable have been used for determining the maximum of time intervals (Δt). The energy and the duration of the first DFDL pulse have been computed at the threshold pump intensity what is necessary to come out the second pulse. Computations have been made at several Δt, and the divergences from exact value (where $\Delta t \rightarrow 0$) have been determined. The results calculated with both explicit and implicit Euler method show it is pay to use the implicit method, and if $\Delta t \leq 3$ ps then both energy and duration of the output pulse approximate the exact value under 1%. For determining the maximum of the space intervals (Δx) a pulse was propagated along the excited volume. The rate of output and input pulse intensity was calculated with both numerical and analitical method. The results calculated with numerical.method have been under 1% if $\Delta x \leq L/550$. According to the limit of the time intervals it's necessary to distribute the length of laser (L = 5.5 mm) into 9, and according to the limit of the space intervals into 550.

IV. NOVEL ALGORITHM

Let the length of laser is distributed into *k* intervals and every intervals are distributed into *m*. The *k* is determined by limit of time intervals and the *m·k* by limit of space intervals. The algorithm has been written down after longer calculations:

$$I^+_{i+m,t} = \xi^m \cdot I^+_{i,t} + \frac{\xi^m - 1}{1-\nu}\left(\nu \cdot 0^- + \frac{\Omega}{\tau(\sigma_o - \sigma_a)}\right),$$

$$I^-_{i-1,t} = \xi^m \cdot I^-_{i+m-1,t} + \frac{\xi^m - 1}{1-\nu}\left(\nu \cdot 0^+ + \frac{\Omega}{\tau(\sigma_o - \sigma_a)}\right).$$

$$\text{where } \xi = 1 + \frac{(\sigma_o - \sigma_a)\cdot \Delta x}{m}(1-\nu)\cdot 0I \;, \quad \nu = \frac{V^2}{4}.$$

$$0I = \frac{N_{i,t} + \ldots + N_{i+m-1,t}}{m}, \quad 0^- = \frac{I^-_{i+1,t} + \ldots + I^-_{i+m,t}}{m},$$

$$0^+ = \frac{I^+_{i-1,t} + \ldots + I^+_{i+m-2,t}}{m}, \quad i = 1,2,\ldots,k\cdot m.$$

The ξ^m is quickly calculated if the $m = 2^j$, where j is an integer number.

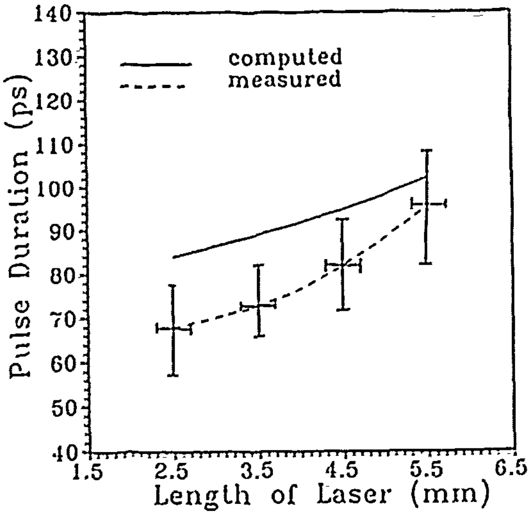


Fig. Measured and calculated pulse duration of the single pulses from a Rhodamine 6G DFDL. The pump intensity was adjusted to the threshold of the second DFDL pulse.

V. RESULTS OF CALCULATIONS

The *k = 10* and *m = 64* has been chosen for the computations. The maesured and the calculated values are shown on the figure. According to the figure the calculated values are in good agreement with the measurements [1].

References:

1. Zs.Bor, A.Müller, B.Rácz, F.P.Schäfer: Appl. Phys. B 27, 9-14 (1982)
2. Zs.Bor, A.Müller, B.Rácz, F.P.Schäfer. Appl. Phys. B 27, 77-81 (1982)
3. Zs.Bor, A.Müller: IEEE J. Quantum Electron. QE-22, 1524-1533 (1986)
4. J.Hebling: Optics Comm. 64, 539-543 (1987)
5. J.Klebniczki, Zs.Bor, G.Szabó: Appl. Phys. B 46, 151-155 (1988)
6. J.Hebling: Appl. Phys. B 47, 267-272 (1988)
7. J.Hebling, J.Seres, Zs.Bor, B.Rácz: Optical and Quantum Electrons 22, 375-384 (1990)
8. H.Kogelnik, C.V.Shank. J. Appl. Phys. 43 (1972)
9. Irl N. Duling III, M.G.Raymer: IEEE J. Quantum Electron. QE-20, 1202-1207 (1984)

ADAPTED VERSIONS OF THE EM ALGORITHM FOR PENALIZED LIKELIHOOD IN EMISSION TOMOGRAPHY

Alvaro R. De Pierro
Institute of Mathematics, Statistics and Computer Science
State University of Campinas
CP 6065, 13081,. Campinas, SP, Brazil

Abstract. We present in this paper two different methods for solving the penalized likelihood maximization problem arising in emission computed tomography (ECT). Both methods are modifications of the Expectation Maximization (EM) algorithm proposed to overcome the inability of this algorithm in its usual form to cope with penalization terms in a non expensive way.

I. INTRODUCTION

In ECT we aim to reconstruct a function that is the distribution of radioactivity in a body cross-section and the measurements are used to estimate the total activity along lines of known location. Higher levels of noise induce the use of mathematical models incorporating the statistical nature of the process instead of inverting the Radon transform as in x-ray computed tomography [1]. In [2], Shepp and Vardi, suggested the use of maximum log-likelihoods estimates derived from the Poisson nature of the emission process, i.e.,

$$\max_{x\geq 0} L(x) = \sum_{i=1}^{m} y_i \ln\langle a_i, x\rangle - \langle a_i, x\rangle, \qquad (1)$$

where $y = \{y_i\}$ $(i = 1, \ldots, m)$ are the photon counts, a_i are the columns of $A = \{a_{ij}\}$, the matrix modelling emission features, and $x = \{x_j\}$ $(j = 1, \ldots, n)$ the image vector (emission density) to be reconstructed ($\langle , \rangle$ denotes the standard inner product and $\sum_{i=1}^{m} a_{ij} = 1$).

To solve (1), Shepp and Vardi proposed to use the EM algorithm obtaining very good results for earlier steps. Unfortunately, iterations have to be stopped before a deteriorating effect (irregular high amplitude patterns) appears. To cope with this inconvenient, quadratic penalization terms have been suggested [3], so problem (1) turns now to be

$$\max_{x\geq 0} L(x) - \frac{\gamma}{2}\, p(x), \qquad (2)$$

where $p(x)$ is a convex quadratic function and γ a positive parameter. For the new problem, the standard EM algorithm is no longer applicable in practice except if the matrix associated with $p(x)$ is diagonal. In the following sections we describe the EM algorithm for (2) and for the pure quadratic problem, as well as the new alternatives proposed for (2).

II. THE EM ALGORITHM

Let Y be a random vector (observed data in some experiment) with density function $g(Y,x)$, where x is some vector of parameters to be estimated. If g is difficult to maximize with respect to x, a possible solution is to embedd Y in a richer sample space X where the optimization problem is easier to solve. Then, the EM algorithm is defined as. given $x^0 \in \Omega$ . .he parameter space)

$$x^{k+1} = \arg\max_{x\in\Omega} q(x/x^k), \text{ for } k=0,1,2,\ldots \qquad (3)$$

where $q(x/x^k)$ is the expectation of the extended log-density given (Y,x^k). Convergence properties and more details for (3) can be found in [4] and [5].

Two examples are of special interest, the penalized log-likelihood (2) and the least squares case. For the first we define the complete data space as the set of independent Poisson distributed variables x_{ij}, interpreted in ECT as the number of emissions in pixel j detected by tube i [2]. So, (3) becomes equivalent to maximize

$$q(x/x^k) = \sum_{j=1}^{n}\sum_{i=1}^{m} \frac{y_i a_{ij} x_j^k}{\langle a_i, x^k\rangle} \ln a_{ij}x_j - a_{ij}x_j - \frac{\gamma}{2}p(x), \qquad (4)$$

taking into account that $E(x_{ij}/Y,x^k) = \dfrac{a_{ij}x_j^k y_i}{\langle a_i, x^k\rangle}$.

If

$$L(x) = \frac{1}{2}\sum\left(b_i - \langle h_i, x\rangle\right)^2 = \frac{1}{2}\, x^t H^t H x - H^t b + \frac{b^t b}{2}, \qquad (5)$$

we consider $b_i = \sum_{j=1}^{n} b_{ij}$, b_{ij} normally distributed $N(h_{ij}x_j,\ n)$ and $\tilde{b}_{ij} = E(b_{ij}\ /\ b, x^k) = h_{ij}x_j{}^k + \frac{1}{n}\left(b_i - \langle h_i, x^k\rangle\right)$ (see[6]). So

$$q(x/x^k) = -\frac{n}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\tilde{b}_{ij} - h_{ij}x_j\right)^2. \qquad (6)$$

and differentiating for $j = 1, \ldots, n$ the equations are

$$-\, n x_j^k \Big(\sum_{i=1}^{m} h_{ij}^2\Big) - \sum_{i=1}^{m} h_{ij}\left(b_i - \langle h_i, x^k\rangle\right) +$$

$$n x_j \Big(\sum_{i=1}^{m} h_{ij}^2\Big) = 0 \qquad (7)$$

III THE EPM ALGORITHM

Let now

$$p(x) = \frac{1}{2}\, x^t S x - x^t q, \qquad (8)$$

where $S = \{s_{ij}\}$ is a positive semidefinite n x n matrix, q an n-vector. If we apply (3-4) to (2) using (8) the system to be solved is for $j = 1, \ldots, n$.

$$\frac{x_j^k}{x_j}\sum_{i=1}^{m}\frac{y_i a_{ij}}{\langle a_i, x^k\rangle} - 1 - \gamma\Big(\sum_{l=1}^{m} s_{jl}x_l - q_j\Big) = 0 \qquad (9)$$

Unless off-diagonal elements of S are zero, (9) is a huge system of nonlinear equations. Several alternatives have been suggested to cope with this drawback ([6] and [7]), but they are not convergent for every γ. Our first alternative is to substitute in (3) Ω by a partial maximization and the new algorithm becomes solving (9) for $j \in Ji_k$ where Ji_k is a subset of indices chosen in such a way that $j \in Ji_k \nRightarrow s_{jl} = 0$ for $l \neq j$, i_k is a control for the sequence and the blocks contain all the variables for each cycle. If S is sparse the subsets J_i are a few number (typically q for a smoothing matrix). In [8] (Theorem 4.1) we proved that the general EPM algorithm converges. Inplementation and experiments can be found in [7].

IV THE EXTENDED EM ALGORITHM

In spite of the advantage of being convergent for every γ the EPM algorithm is quite expensive because of the up-dating of the scalar products (see [7]). So we propose an alternative based on introducing a set of artifical data as described next.

$S = H^tH$ for some $H \in R^{nXn}$; on the other hand if p has a minimum (standard assumption), $Sx = q$ has a solution and $q \in R\ (H^t)$, i.e., $q = H^tb$ for some b. Formally we can think that b is a ncrmally distributed random vector and the same procedure as for (5) is applicable. Applying the expectation, first with respect to y, afterwards with respect to b, gives (combining (4) and (6)) =

$$q(x/x^k)=\sum_{j=1}^{n}\sum_{i=1}^{m}\frac{y_i a_{ij} x^k}{<a_i,x^k>}\ln a_{ij}x_j - a_{ij}x_j - \frac{n}{2}\gamma\sum_{i,j}(\tilde{b}_{ij}-h_{ij}x_j)^2 \qquad (10)$$

Differentiating, x_j^{k+1} will be the unique positive solution of (for $j = 1, \ldots, n$)

$$\frac{x_j^k}{x_j}\sum_{i=1}^{m}\frac{a_{ij}y_i}{<a_i,x^k>} - 1 - n\gamma s_{ij}\ x_j^k - \gamma(q_j - <s_j,x^k>) + n\ \gamma\ s_{jj}\ x_j = 0. \qquad (11)$$

taking into account that $s_{jj} = \sum_{i=1}^{m} h_{ij}^2$ and $q_j - <s_j,x^k> = = \sum_{i=1}^{m} h_{ij}(\tilde{b}_i - <h_i,x^k>)$.

Now variables are separated and (11) is a single unknown quadratic equation for each j. Convergence of the algorithm is guaranteed because it is a special case of the EM algorithm

Further work has to be done in order to test practical performance of this algorithm as well as extensions to other nonquadratic penalizations.

References.

1. G.T. Herman, Image Reconstruction from Projections: The Fundamentals of Computerized Tomography, Academic Press, New York, 1980.

2. L.A. Shepp and Y. Vardi, "Maximum likelihood reconstruction in positron emission tomography", IEEE Trans.Med. Imaging, MI-1 pp. 113-122, 1982.

3. E. Levitan and G.T. Herman, "A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography". IEEE Trans.Med. Imaging, MI-6, pp 185-192, 1987.

4. Y. Vardi, L.A. Shepp and L. Kaufman, "A statistical model for positron tomography", J. Amer. Statis. Assoc., 80 pp 8-35, 1985.

5. I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures", Technical Report, Mathematical Institute of the Hungarian Academy of Sciences.

6. P. Green, "On the use of the EM algorithm for penalized likelihood estimation", J.R. Statist.Soc. B, 523, pp 443-452, 1990.

7. G.T. Herman, D. Odhner, K.D. Toennies and S.A. Zenios, "A parallelized algorithm for image reconstruction from noisy projections" Technical Report MIPG 155, U. of Pennsylvania, 1989.

8. A.R. De Pierro, "A generalization of the EM algorithm for maximum likelihood estimates from incomplete data", Technical Report MIPC 119, U. of Pennsylvania, 1987.

BOUNDARY ELEMENT METHOD FOR NONLINEAR ELLIPTIC PROBLEMS WITH NONLINEAR MATERIAL CONDITIONS

K. RUOTSALAINEN
University of Oulu,
Faculty of Technology, Section of mathematics,
SF-90570, Oulu, Finland

Abstract. We shall consider the numerical analysis of the boundary element for nonlinear elliptic problems with nonlinear material conditions. The convergence of the Galerkin and collocation schemes is proved.

1. Introduction

In this paper we study the possibility to apply the boundary element methods to nonlinear elliptic boundary value problems with nonlinear differential equation. To this end in the analysis of the nonlinear boundary element method one has frequently restricted to the cases where the differential equations are linear. The nonlinearity appears only in the boundary conditions.

By means of the Kirchhoff transform we are able to linearize the differential equation. The introduced new unknown function satisfies a linear differential equation. The boundary conditions are, however, nonlinear. This can be done when the the differential operator is in the divergence form and the nonlinearity depends only on the function itself; not on its derivatives.

By the indirect approach of the boundary integral equation methods the "linearized" problem can be transformed to nonlinear boundary integral equation for the unknown boundary distribution which is to be solved numerically. We analyse both the Galerkin and collocation methods for finding an approximate solution. Here we present the convergence results and some preliminary error estimates. The complete analysis is presented in the forthcoming paper [4].

2. The formulation of the problem

We shall consider a nonlinear boundary value problem that is to be encountered in a stationary heat conduction problem with a temperature dependent heat conductivity. The problem consists in finding a potential function $\phi \in W^{1,2}(\Omega)$ such that

$$\begin{aligned} -\nabla \cdot (a(\phi)\nabla\phi) &= 0, \qquad \text{in } \Omega \\ -a(\phi)\frac{\partial\phi}{\partial n}|_\Gamma &= G(\phi|_\Gamma) - f, \quad \text{on } \Gamma. \end{aligned} \tag{1}$$

Throughout the paper we assume that Ω is a bounded plane domain with a regular boundary Γ. In other words the boundary Γ has a regular parameter representation $x : \mathrm{R} \to \Gamma$ with a nonvanishing Jacobian: $\frac{dx}{d\xi} \neq 0$. The symbol $\frac{\partial}{\partial n}$ stands for the outer normal derivative as usual.

Before proceeding with the reformulation of the problem we recall that $W^{m,p}(\Omega), 1 < p < \infty$, is the usual Sobolev space with usual norm $\|u\|_{m,p}$. Besides these spaces we need in the sequel the Sobolev-Slobodetckii spaces $W^{r,p}(\Gamma)$ on the boundary. Other related function spaces are introduced in the order of occurence.

Let us now consider the reformulation of the problem (1). For that we shall make the following basic assumption: We suppose that $a(\phi)$ is a sufficiently smooth function and that there exists positive constants m and M such that for all $\phi \in \mathrm{R}$

$$0 < m \leq a(\phi) \leq M < \infty. \tag{2}$$

Now for the Kirchhoff transform defined by setting

$$K(\phi) = \int_0^\phi a(s)\,ds$$

it holds [4]:

Lemma 1. *Let the function space X be $L^2(\Omega)$ or $L^2(\Gamma)$, respectively. Then the Kirchhoff transform $K : X \to X$ is Lipschitz-continuous and strongly monotone, i.e. for every $u, v \in X$*

$$(K(u) - K(v), u - v)_X \geq m\|u - v\|^2.$$

Furthermore. The mapping $K . W^{1,2}(\Omega) \to W^{1,2}(\Omega)$ is bijective.

By the theory of monotone opretors we easily conclude that the inverse transform $K^{-1} : L^2(\Gamma) \to L^2(\Gamma)$ is also a strongly monotone and Lipschitz-continuous mapping. With the previous lemma it is easy to verify that $\psi = K(\phi)$ satisfies the following potential problem

$$\begin{aligned} \Delta\psi &= 0, \qquad \text{in } \Omega. \\ -\frac{\partial\psi}{\partial n} &= G(K^{-1}(\psi)) - f, \quad \text{on } \Gamma. \end{aligned} \tag{3}$$

provided ϕ solves the problem (1). The converse also holds if ψ is the solution of our original problem (1). This is to say that problems (1) and (3) are equivalent in the weak form.

The problem (3) can be formulated as a nonlinear boundary integral equation [3]. This can be accomplished by introducing a boundary distribution u such that

$$\psi(x) = -\frac{1}{2\pi}\int_\Gamma u(y)\log|x - y|\,ds_y, x \in \Omega.$$

Then by the trace properties of the normal derivative of the single layer potential we derive the nonlinear boundary integral equation

$$A(u) = (\frac{1}{2}I - D^*)u + G(K^{-1}(S(u))) = f, \tag{4}$$

where the operator D^* is the spatial adjoint of the double layer operator D, which is defined by setting

$$Du(x) = \frac{1}{2\pi}\int_\Gamma u(y)\partial_n \log|x - y|\,ds_y.$$

and S denotes the single layer operator defined as

$$Su(x) = -\frac{1}{2\pi}\int_\Gamma u(y)\log|x - y|\,ds_y.$$

For the unique solvability of the boundary integral equation (and of the boundary value problem , we make the following assumption

A1. *The nonlinear operator $G : L^p(\Gamma) \to L^q(\Gamma), 2 \leq p < \infty, \frac{1}{p} + \frac{1}{q} = 1$, is bounded, continuous and strictly monotone. In addition to these we suppose that for almost all $x \in \Gamma$*

$$G(u)(x) \geq \alpha|u(x)|^p + h, \alpha > 0, h \in \mathrm{R}.$$

Then the boundary integral equation is uniquely solvable. More precisely we have [3],[4]:

Theorem 2. *For every $f \in L^q(\Gamma)$ there exists a unique solution $u \in L^q(\Gamma)$ to (4).*

3. The boundary element discretization

As the approximation schemes we shall use the Galerkin and collocation methods. For the approximation we use the boundary element spaces $S_N^d(\Theta)$ with respect to the partition $\Theta = \{x_i = x(\tau_i) | i = 0, \ldots, N-1\}$ on Γ, where $\{\tau_i\}$ are grid points on the unit interval that is carried to Γ by the parameter representation. We assume that the family of the partitions is quasiuniform. We remind the reader that $S_N^d(\Theta)$ corresponds via the parameter representation the smoothest 1-periodic splines of degree d on the unit interval [1]. As a mesh parameter we choose $h = \frac{1}{N}$.

The problem, in general, is to find the coefficients $\alpha_i \in \mathbb{R}$, $i = 0, \ldots, N-1$ such that $u_h = \sum_{i=0}^{N-1} \alpha_i \psi_i$ is as good approximation of the true solution as possible. The functions ψ_i forms a suitable basis of the spline space. We shall present here the convergence analysis of the two most popular methods.

The Galerkin method. As it is well-known in the Galerkin method the coefficients are fixed by means of the orthogonality condition

$$(A(u_h), \varphi)_{L^2(\Gamma)} = (A(u), \varphi)_{L^2(\Gamma)}, \ \varphi \in S_N^d(\Theta) \qquad (5)$$

This is the same as to treat the family of the operator equations

$$(\frac{1}{2}I + P_h D^*)u_h + P_h G(K^{-1}(S(u_h))) = P_h f, \qquad (6)$$

where $P_h : L^2(\Gamma) \to S_N^d(\Theta)$ is the orthogonal projection

In the convergence analysis we apply the theory of a-proper-mappings [2]. Since the nonlinear mapping $A(\cdot)$ is an operator of Gårding type we have [4]:

Theorem 3. *The nonlinear operator $A(\cdot) : L^q(\Gamma) \to L^q(\Gamma)$ is a-proper with respect to the projectionally complete scheme $\{P_h, S_N^d(\Gamma)\}$.*

For the solvability of the Galerkin equations we shall need the following assumption, which usually valid in true applications.

A2. *$G : L^p(\Gamma) \to L^q(\Gamma)$ is continuously Frèchet-differentiable, and that the Fréchet-derivative $DG(u)$ is strictly monotone (i.e. positive).*

With this assumption the nonlinear operator has the following properties:

Theorem 4. *The Fréchet-derivative $DA(u) : L^q(\Gamma) \to L^q(\Gamma)$ is a Fredholm operator with vanishing index. $ind(DA(u)) = 0$ in addition to this the derivative is one-to-one.*

As a corollary we conclude [4]:

Theorem 5. *There exists $h_0 > 0$ such that for every $0 < h < h_0$ the Galerkin method yields a unique solution to (5) (or (6)), and $\|u - u_h\| \to 0$ as $h \to 0$.*

Finally, utilizing the Fredholmness of nonlinear operator $A(\cdot)$ and the approximation properties of spline spaces on Γ we are able to derive the asymptotic error estimates:

Theorem 6. *Let u_h be as in the previous theorem. Then for sufficiently small h there holds*

$$\|u - u_h\|_{W^{t,q}(\Gamma)} \leq c(\|u\|_{L^q(\Gamma)})\, h^{s-t}\, \|u\|_{W^{s,q}(\Gamma)}$$

for every $-1 \leq t \leq 0 \leq s \leq d+1$.

The collocation method. For the collocation method we define the collocation points as follows.

$$\tilde{x}_i = x_i, \quad d \text{ is odd},$$

$$\tilde{x}_i = x(\frac{\tau_i + \tau_{i+1}}{2}), \quad d \text{ is even}.$$

Here the points τ_i on the real line corresponds the grid points of the partition Θ via the parametrization of Γ. If we let I_h to denote the interpolation operator, which interpolates between the function values on the collocation points $\tilde{x}_i$, from the space of continuous functions to the approriate spline space $S_N^d(\Theta)$, we can write the collocation equations in the form

$$I_h A(u_h) = I_h f. \qquad (7)$$

The interpolation equation (7) is not always reasonable. Therefore we must have some additional properties besides the asssumptions A1 and A2. We require that

A3. *The mapping $\bar{G} : W^{\frac{1}{2}+\epsilon,p}(\Gamma) \to W^{\frac{1}{2}+\epsilon,q}(\Gamma)$ is bounded with some ϵ that is sufficiently small ($0 < \epsilon < \frac{1}{2}$).*

After this additional property the collocation equations (7) make sense. The more severe difficulty in the convergence analysis is due the fact that $\{I_h, S_N^d(\Theta)\}$ is not a projectionally complete scheme in $L^q(\Gamma)$. However, since the collocation equations can be written as

$$(\frac{1}{2}I - I_h D^*)u_h + I_h G(K^{-1}(S(u_h))) = I_h f$$

we don't need the interpolation operator to be bounded in $L^q(\Gamma)$. This is due the fact that the double layer and single layer operators are smooth operators. This allows us to extend the proof of the convergence of the Galerkin method to the collocation method. Thus we have [4]:

Theorem 7. *There exists a mesh parameter h_0 such that for all $0 < h < h_0$ the collocation equations admit a unique solution and $\|u - u_h\|_{L^q(\Gamma)} \to 0$ as $h \to 0$.*

References

1. D.N. Arnold and W.L. Wendland, *The convergence of spline collocation for strongly elliptic equations on curves*, Num. Math. 47 (1985), 317–343.
2. W.V. Petryshyn, *On the approximation-solvability of equations involving a-proper and pseudo-a-proper-mappings*, Bull. A.M.S. 81 (1975), 223–312.
3. K. Ruotsalainen, *Remarks on the boundary element method for strongly nonlinear problems*, Applied Mathematics Preprint AM90/11 (1990), The university of New South Wales, Australia.
4. K. Ruotsalainen, *The convergence of the boundary element methods for nonlinear elliptic problems* (to appear).
5. K. Ruotsalainen and W.L. Wendland, *On the boundary element method for some nonlinear boundary value problems*, Num. Math. 53 (1988), 299–314.

A Conceptual Architecture for Modelling Physical Systems

C. P. McGann and J.B. Grimson
Dept. of Computer Science,
Trinity College,
University of Dublin,
Dublin 2,
Ireland.

Tel: +353-1-772941
Fax: +353-1-772204
Email: CPMCGANN@VAX1.TCD.IE

D.P. Finn
Hitachi Dublin Laboratory,
Hitachi Europe Ltd.,
O' Reilly Institute,
Trinity College,
University of Dublin,
Dublin 2,
Ireland.

Tel: +353-1-6798911
Fax: +353-1-6798926
Email: DFINN@VAX1.TCD.IE

Abstract

A conceptual architecture for modelling abstract physical systems is described. The architecture aims to provide a system framework which is based on the engineering modelling process. This paper concentrates on the use of qualitative causal networks as a tool for predicting and estimating the physical phenomena acting within a physical system. Three qualitative simulation techniques are assessed and a constraint-based qualitative simulation approach is adopted. The incorporation of this approach within the overall system architecture is discussed.

1. Introduction

1.1 Background

Modelling of engineering problems can be divided into a number of stages; geometric modelling, physical modelling, mathematical modelling, numerical modelling and graphics modelling Geometric modelling involves representing the geometric features of real world problems and if possible making geometric simplifications to reduce the complexity of the problem. Physical modelling involves identifying any physical phenomena which are occurring such as heat transfer, fluid flow or stress and if possible, making certain assumptions to simplify phenomena complexities. Mathematical modelling requires building suitable equations to represent the problem mathematically and selecting correct boundary conditions Numerical modelling involves constructing suitable numerical algorithms and solving these algorithms computationally Finally graphical modelling requires the use graphical techniques to present the numerical solution. Many existing numerical problem solving environments such as DEQSOL [Umetani *et al.* 1985, Kon'no *et al.* 1986], ELLPACK [Rice 1985] and FIDISOL [Schonauer and Schnepf 1987], NEXUS [Gafney *et al.* 1986] help users with the mathematical and numerical modelling stages of the overall modelling process. Few systems, however, have focussed on the initial modelling stages, namely, geometric and physical modelling In these modelling stages, the user is generally required to conceptually represent a real world problem, identify the nature and relative importance of the physical phenomena occurring and determine any geometric simplifications or phenomena assumptions which may be made to facilitate efficient numerical analysis.

1.2 Modelling of Physical Systems

This paper aims to address these shortcomings by presenting a conceptual architecture which addresses the process of engineering modelling for the problem domain described by partial differential equations. The discussion assumes as a starting point, a high-level, abstract, three dimensional representation of a real world problem or physical system (referred to as the complex model). This representation then evolves through a series of transition processes, each stage representing a simpler model. The final model (called the optimised model) is suitable for mathematical and numerical analysis The architecture includes techniques for representing real world problems, identifying all physical phenomena and qualitatively simulating their behaviour. Additionally, the user may select significant phenomena for further analysis, carry out reduction of geometric and phenomena features to evolve an optimised geometry which is suitable for efficient mathematical and numerical modelling Although all modelling stages are initially discussed in the paper, special emphasis is placed on the behavioural prediction of the physical phenomena. In particular, the integration of the two important artificial intelligence techniques of qualitative reasoning and causal network analysis are focused on to establish a high-level, knowledge intensive network representation

The paper is divided as follows; Section 2 gives an overview of the modelling process. Section 3 discusses other research which has addressed these higher-level stages of engineering modelling. The relevance of the proposed modelling methodology is illustrated and the incorporation of certain current research into the proposed architecture is discussed. Section 4 deals with qualitative reasoning as a key inference technique and discusses three principal qualitative reasoning methodologies. Section 5 discusses the fundamental approach taken in this work, i.e., qualitative causal networks. Section 6 concludes the paper.

2. System Outline

In this Section an overview of the important stages of the modelling process in the proposed architecture is presented. Furthermore, the intermediate stages of the modelling process are outlined. Having established a global perspective on the system architecture, system entities are introduced which form a framework for the modelling process.

2.1 System Architecture

Figure 1 gives an overview of the system architecture. Each stage depicts an evolution of the physical problem, beginning with the *complex model* at the highest level of abstraction and culminating in an *optimised model* which corresponds the lowest representation level before mathematical modelling. Transition processes determine each evolution.

(i) Stage 1: 3-D Representation
The *complex model* is created by the user and is a representation of a real world problem or physical system as shown in Figure 2. The complex model consists of a number of components. Each component is represented as an object with initial attributes of geometry, function, location and material. At this stage the user is requested to specify any known phenomena features for each component. In the example illustrated in Figure 2, the user may be asked if an electric current, voltage difference or electric field are present.

(ii) Transition A: Cause-Effect Inference
A shallow rule based inference mechanism is used to exhaustively predict all possible phenomena in the system based on the high-level knowledge available from the 3-D representation.

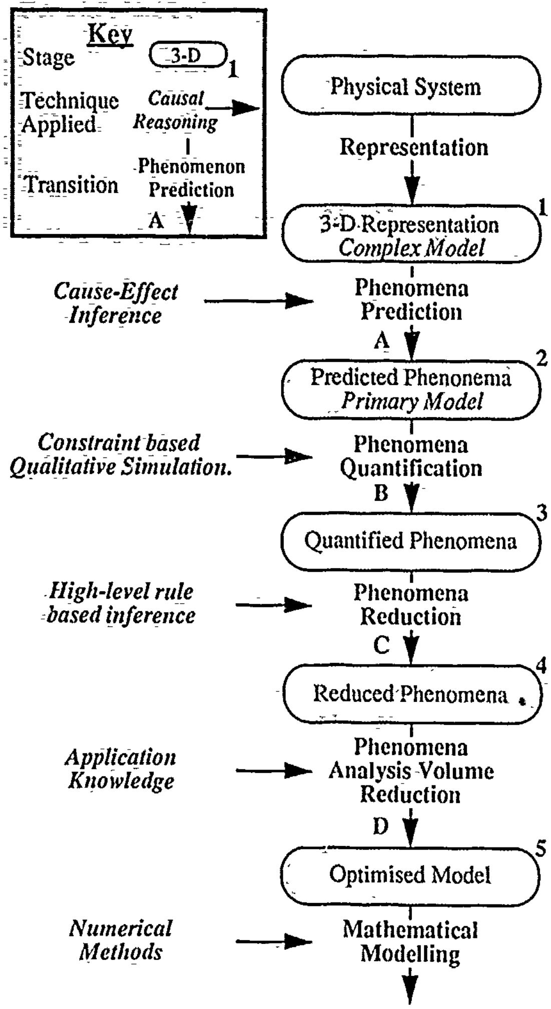


Figure 1 Overview of proposed modelling architecture

(iii) Stage 2: Predicted Phenomena
At this stage, all known phenomena have been predicted by the system. However, it may not be realistic to analyse all phenomena, therefore by quantifying each phenomenon, certain phenomena may be found to be insignificant and therefore removed from further consideration.

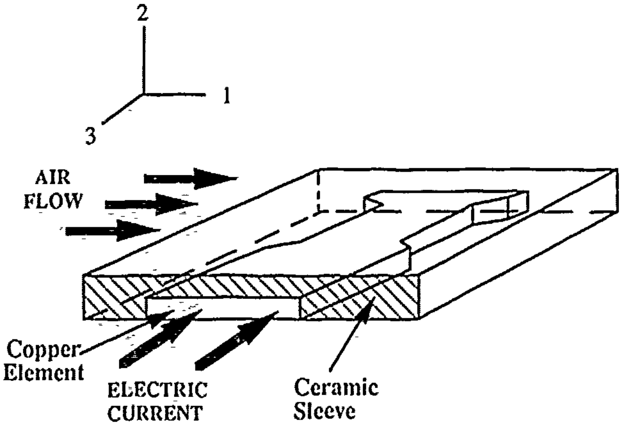


Figure 2 Physical System or Real World Problem

(iv) Transition B: Constraint-based Qualitative Reasoning
Qualitative reasoning and *causal networks* are used to derive a *behavioural description.* Any simplifications such as *geometric idealisations* are carried out to derive a simplified primary model for each phenomenon, these simplifications are known as primary simplifications. Each phenomena is then estimated using constraint-based qualitative reasoning to give a behavioural description. This description can then be assessed by the user to estimate the importance of each phenomenon. This approach is discussed in detail in Section 4.

(v) Stage 3: Quantified Phenomena
At this stage there is a stable description of the behaviour of all components of the simplified complex model.

(vi) Transition C: Phenomena Reduction
Shallow rule-based inference mechanisms are used in this transition to examine the values of key parameters for each phenomena throughout the system. Using knowledge bases, the relative importance of each phenomenon can be assessed, e.g., stress levels in electronic components and the user may make an informed selection of phenomena to include in subsequent analysis.

(vii) Stage 4: Reduced Phenomena
Stage 4 consists of a number of reduced phenomena for further analysis and optimisation.

(viii) Transition D: Phenomenon and Control Volume Reduction
Depending on the phenomena selected by the user for analyses, an *optimised model* is derived. Optimisation may consist of further geometric simplifications, selection of a subsection of the model based on symmetry or selection of a reduced analysis volume.

(ix) Stage 5: Optimised Model
This final stage is a model of the primary model optimised for subsequent analysis. It reflects a trade-off between simplicity and accuracy in analysis. For example, if the optimised model was to be used for numerical simulation, an important concern would be the scope of the analysis volume. If a large analysis volume is assumed, then the complexity of calculations is increased, whereas if the analysis volume is limited, important detail may be lost.

2.2 Introducing System Entities

The physical system is comprised of a number of components. Each component has a functional perspective and a geometric perspective. The functional perspective determines what use the component has. For example, a metallic pipe may be viewed as a flow channel for a fluid, an electrical conductor or a bearing for a rod that can rotate inside it. The geometric perspective defines the shape and dimensions of a component. Initially a three dimensional shape may be selected by the user from a limited library of shapes and instantiated with the appropriate dimensions e.g. a BLOCK of length A, width B and depth C. The geometric perspective evolves from the complex model through the primary simplification to the optimised model.

Each component has a behaviour. The behaviour of that component is essentially the physical phenomena occurring in that component and is context dependent. The context of a component determines the component's behaviour and it is determined by the component's geometric perspective, functional perspective and boundary links as well as its current local behaviour.

Boundary links are the medium through which effects are propagated between two connected components. A valid boundary exists between any two physically connected components and also between a component and the outside world, where they are directly in contact. A boundary link is an abstract entity which models the transfer of effects of physical phenomena between neighbouring components. Transfer parameters are defined for effects of physical phenomena, e.g., for heat transfer define 'Heat Flux' or for a stress field define 'Force'. This abstraction enforces the localised context for each component. For a component producing a heat field, it need only output a 'Heat Efflux' parameter value to its boundaries. It is up to each boundary to present this effect to the context of the neighbouring component in a suitable form. Thus the context of a neighbour will incorporate 'Heat Flux' which will be presented as a

'Heat Influx' parameter with a set value. From the receiving components perspective, the origin of the heat is immaterial as are the destinations for subsequent side-effects. A boundary link will be represented as an object with initial attributes of shape and participating components.

3. Related Work

While there has been considerable investigation of different types of abstract models for physical systems, the problem of modelling physical phenomena in such systems has received little attention. However, work has been done which is related to parts of the proposed modelling process.

Several artificial intelligence researchers have investigated the use of qualitative reasoning to predict the behaviour of physical systems. Section 4 explores the contribution of this work in greater detail. The proposed system is a modelling process directed to modelling physical phenomena whereas the applications of existing qualitative reasoning systems have focussed on predicting device behaviour, e.g., [Forbus et al '87].

Geometric idealisation plays an important role in engineering modelling. [Baehman et al '88], [Baehmann '88] and [Collar '90] have done considerable work in the field of geometric modelling but the starting point for their idealization process is a lower level, fully specified physical model. In contrast, the proposed system begins with a high-level, incomplete, abstract specification (i.e. the complex model) where the behaviour of the system is not yet determined. Nonetheless, the geometric idealization techniques proposed in their research is valid in the lower levels of the proposed architecture specifically at phenomena and analysis volume reduction stages.

Context dependent behaviours [Nayak et al '90] represent a conceptually appealing method for organising multiple models of primitive components. Unfortunately, like component based and process based qualitative reasoning systems, this ontology is geared towards prediction of device behaviour and has limited application in modelling physical phenomena.

[Gelsey '90] describes a program for automated physical modelling. He proposes a quantitative modelling approach whereby the behaviour of the system (a mechanical device) is derived using numerical simulation. Again, the application for this work is in the area of kinematic analysis rather than physical phenomena. Furthermore, the use of quantitative techniques requires a more complete model specification than is provided for at the user interface of the proposed system.

4. Qualitative Reasoning

Most of today's expert systems have invariably modelled their domain with a "black box" approach, i.e., compiling rules from observable inputs and outputs, with no consideration for underlying physical mechanisms. A commonly recognised failing of such "shallow models" is that they are highly domain specific [Kuipers '86]. The black box approach to knowledge representation is a shallow way of contriving summary rules to suit the needs of the application which must use them.

Qualitative reasoning provides for "deeper" knowledge which models the underlying systems of the black box and allows the derivation of the high level rules of input-output by applying an input to the model and allowing simulation to predict the output. In contrast to shallower conventional heuristic models, a qualitative reasoning system will embed its knowledge in real physical mechanisms rather than in rules of thumb based on intuition and experience.

In this section three approaches to qualitative reasoning are discussed. In particular abstract qualitative reasoning systems are defined and how principal components are instantiated within component based, process based and constraint based systems respectively. Furthermore, the relevance of these methodologies in the context of the proposed architecture is discussed (Figure 1 Transition B).

4.1 Components of a qualitative reasoning system.

A qualitative reasoning method begins with a model of the domain, called a *structural description*, in which precise numerical values and precise functional relationships are absent. All parameters can take on one of a finite set of non-numeric values which generally delimit regions of qualitatively distinct behaviour (e.g., fluid flow rate could be expressed as being laminar, between laminar and turbulent, turbulent). An ordered set of such qualitative values is called a *quantity space*. The minimal quantity space has the values (, 0, +) which only supports a system to reason about signs.

The influence of one parameter on another is expressed by *qualitative constraint equations*. The qualitative state of the system is an assignment of a qualitative value and *Incremental Qualitative* value or IQ value (i.e., direction of change which may be specified as decreasing, steady, or increasing) to each parameter together with the time-point or interval at, or over, which this value applies and the direction in which the parameter value is changing (if at all). Time is represented as an ordered sequence of time-points generated dynamically whenever something interesting happens to a parameter Transition rules govern any changes in parameter value.

Some systems based on qualitative reasoning use simulation as their inference procedure, while others use envisionment Simulation begins with an initial state description and propagates values forward, exploring all legal possibilities, until no more transition rules apply. Envisionment generates all legal qualitative states and then all possible transitions between these states Both approaches result in a graph of possible states with arcs signifying permissible transitions between states. A qualitative behavioural description is then any path through such a graph.

So any qualitative reasoning system can be characterised by its *structural description, quantity space, notion of qualitative state, representation of time, transition rules, behavioural description* and *inference procedure*. The major difference between the various approaches to qualitative reasoning concerns the precise statement of the constraint laws and how they are derived from the physical structure [de Kleer and Brown '83] A central organising principle for de Kleer and Brown [de Kleer and Brown '84] is the notion of component, for Forbus the notion of process [Forbus '84] and for Kuipers the notion of constraint [Kuipers '86]. The preceding views of parameters, constraints, qualitative state and representation of time are essentially shared among the qualitative physics of component based, process based and constraint based systems.

4.2 Envisioning

De Kleer and Brown take the view that a device consists of physically distinct parts connected together. The goal is to draw inferences about the behaviour of the composite device solely from laws governing the behaviours of parts. Their central modelling primitive is the qualitative differential equation, called a confluence, which acts as a constraint on the variables and derivatives associated with components. For example, consider the qualitative behaviour of a pressure regulator (Figure 3) expressed by dP+dA-dQ=0, where P is the pressure across the valve, Q is the flow through the valve and A is the area available for flow. dP, dQ and dA represent the changes in P, Q and A respectively. The confluence represents multiple competing influences. the change in area positively influences flow rate and negatively influences pressure The change in pressure positively influences flow rate, etc. A confluence is generally valid for a certain operating range of some component In this example very different behaviours occur when the valve is fully open or fully closed.

4.3 Qualitative Process Theory

Forbus describes a physical situation in terms of the interaction of competing processes [Forbus '84]. The central idea is that all changes in physical systems are caused directly or indirectly by processes. Processes can become active or inactive as parameter

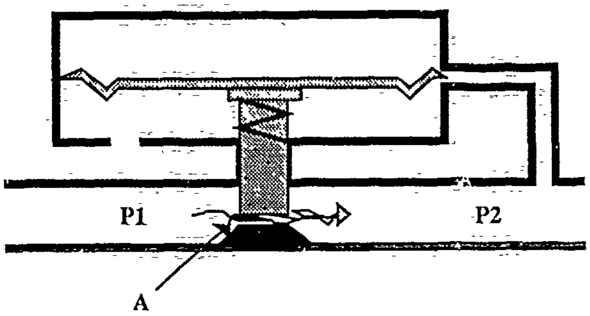


Figure 3 Pressure Valve Example

values change by the action of other processes. The rules governing a process indicate:

1. Under what conditions a process holds, e.g., the temperature of the source must exceed the temperature of the destination for heat flow to occur.
2. The relations it imposes among parameters.
3. The influences it imposes on the parameters (e.g., the amount of heat at the source is negatively influenced by the flow rate of heat leaving it).

The physical situation presented in Figure 4 can be described by the heat-flow process. The quantity condition for the heat-flow process is that the temperature of the source is greater than the temperature of the destination of the heat The heat-flow rate negatively influences the heat of the source and positively influences the heat of the destination. The complete constraint on the amount of heat in the source is determined by the sum of all the influences which reference it.

4.4 Qualitative Simulation

In Kuiper's constraint based approach, the constraints on how parameters are related to each other are two- or three-place relations on physical parameters Some specify familiar mathematical relationships: DERIV (velocity, acceleration), MULT (mass, acceleration, force), MINUS (forward, reverse). Others assert qualitatively that there is a functional relationship between two physical parameters, but only specify that the relationship is monotomically increasing or decreasing. M+(age,experience), M-(mpg,mph). Inequality and conditional constraints specify conditions under which some constraint holds.

The "causal structure description" indicates each of the constraints and parameters of the model. Consider the simple physical system in Figure 4 consisting of a closed container of gas (at temperature T) that receives heat from a source (Ts) and radiates heat into the air (Ta). The rate of flow of heat into the gas is a strictly increasing function of the temperature difference between the gas and the source. dT = 0 corresponds to no heat flow into the gas. To solve this model, a qualitative simulation is carried out by propagating +,0,- values (using Figure 5) and inequalities (using constraint propagation) in order to obtain values for all the variables.

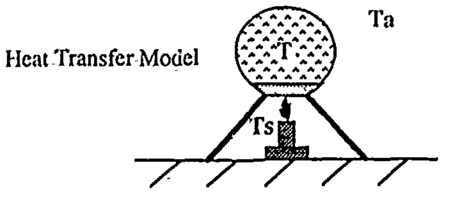


Figure 4 Heat Transfer Example

4.5 Qualitative Reasoning for Modelling Physical Phenomena

De Kleer and Brown's component based reasoning system determines a composite device behaviour from its component structure. Domains consist of specific components which have their structural description mapped out explicitly in the form of confluences (qualitative differential equations). The focus for behaviour prediction rests on the behaviour of the *device* itself rather than on underlying internal effects.

A similar observation applies to Forbus' qualitative process theory. The central idea of a process as the essential agent for change in a physical system fits well with the notion of physical phenomena introducing side-effects in other components. However, this approach is also applied to predicting *device* behaviour [Forbus et al '87].

4.6 Proposed Modelling Approach

The modelling approach proposed in this work deals with behaviour at a lower level than component and process based systems. Instead of predicting a composite device behaviour, this work is more concerned with the behaviour *within* a component, i.e., physical phenomena. Rather than a component having a behaviour itself, in the approach taken here, it serves more as a site for behaviours to occur. The composite behaviour in the modelling approach of the proposed system looks at how physical phenomena, local to one site (component) initially, might propagate to other sites (components) in the system.

Kuipers' constraint based approach conforms most readily to the requirements of this work. Constraint equations can be adapted to qualitatively reflect fundamental laws of physics, e.g., power dissipation = M+(voltage). The calculus of qualitative simulation will allow side-effects of physical phenomena to be derived. Parameters provide an ideal method for propagating effects by parameter sharing between neighbouring components.

In contrast to all three qualitative reasoning systems outlined in this section, it's not of interest to predict system behaviour over time. Instead, the proposed system incorporates the qualitative calculus and constraint conventions of Kuipers' simulation system to perform dynamic simulation but present the resulting behavioural description as a static view of the final time-point parameter values. The graph of transitions may be used as a historical account of the simulation to justify results to the user.

5. Qualitative Causal Networks

Reason locally and propagate globally: this is the basic principle from which the qualitative causal network structure is derived as a tool for modelling physical systems. In this section the use of qualitative simulation is explained to determine local behaviour at each component (site) in the network and how a causal network, implemented through boundary links between neighbouring components, is used to propagate side-effects to 'connected' components.

The simulation process described in Section 5.1 corresponds to A and B of Figure 6 which outlines the algorithm used in the proposed system to generate a behavioural description of the physical system (Figure 1, Stage 3). Section 5.2 examines how causal networks are used to propagate side-effect phenomena to the contexts of neighbouring components (Figure 6, Part C).

X:

Y:	-	0	+
-	-	+	?
0	-	0	+
+	?	+	+

Figure 5 Constraint Propogation Table

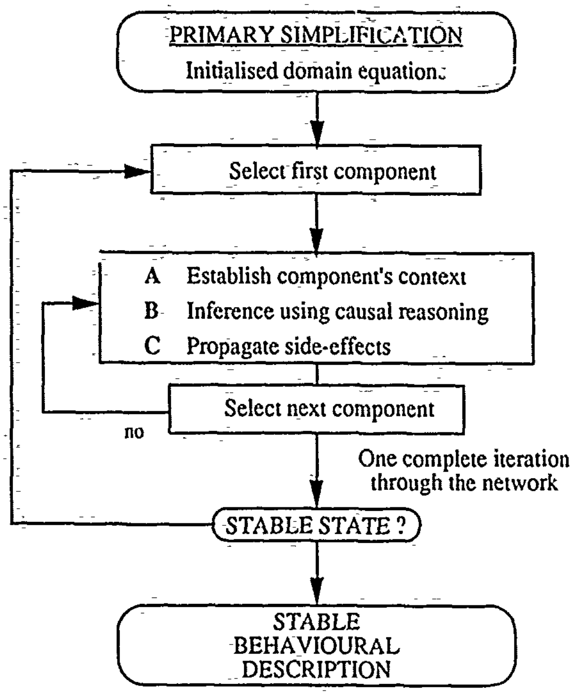

PRIMARY SIMPLIFICATION
Initialised domain equations
Select first component
A Establish component's context
B Inference using causal reasoning
C Propagate side-effects
Select next component
no
One complete iteration through the network
STABLE STATE ?
STABLE BEHAVIOURAL DESCRIPTION

Figure 6 Generating a Behavioural Description

5.1 Localised qualitative simulation

The strategy used in this work is to represent essential equations of physics qualitatively through constraint-equations in the format proposed in Kuipers' qualitative simulation methodology [Kuipers '86], e.g., Bernouilli's equation, universal gas equation etc The knowledge base for the proposed system is divided into a number of domains. Each domain corresponds to a physical phenomenon and the knowledge for that domain will consist of constraint-equations derived from the appropriate laws of physics, e.g., Ohm's law could be represented as. Voltage - MULT(current, resistance) = 0 Qualitative calculus [Kuipers '86] is used to solve these equations

Simulation at a local site begins with an initial state or structural description (Figure 1: Stage 2). In the first iteration, high-level phenomena prediction establishes the domain equations to be included for each component and initialises certain system parameters based on user input (Figure 1. Transition A). Subsequent iterations derive their structural descriptions from their context.

All active parameters (i.e. those parameters with IQ value <> steady) are placed on an active list. Prediction rules are applied to each parameter on the active list to predict all possible transitions with no regard at this stage for the validity of the new value. Transitions are subsequently filtered by applying constraint equations which filter out inconsistent transitions, e.g., if acceleration is constant then, because acceleration and velocity are related by the derivative DERIV(velocity, acceleration), a velocity IQ value of 'steady' would be inconsistent and hence filtered out. Other parameters that had been steady may be perturbed by this process, i.e., a side-effect. When all transitions have been verified or removed, the next state is determined for the behaviour of that component. Side-effects are incorporated into the components boundary link parameters and propagated by the causal network.

5.2 Context building with causal networks

The structural description for qualitative simulation at each node is derived for each component solely from its context. The role of the causal network is to ensure that any side-effect phenomena due to other component behaviours are included in this context.

The basis of the network structure is the boundary links between neighbouring physically connected components. The boundary is characterised by its participant components, its geometry (which may determine the type of distribution for a phenomenon), and two lists of transfer parameters, (i.e., source A <-> destination B and source B <-> destination A).

After a local simulation has been completed, the boundary link's outgoing transfer parameters are updated either by appending a new parameter or by changing a current transfer parameter's value and/or IQ value. The next component incorporates all incoming transfer parameter lists into its context. Simulation then proceeds with the context as an initial state.

5.3 Stopping Criteria

From Figure 6 it is clear that the simulation process is an iterative one. One complete iteration corresponds to a local simulation/global propagation cycle. The behavioural description is considered to be complete when the system reaches an equilibrium. In practice, a physical equilibrium will be represented by a complete iteration which produces no new states of parameters. Each component's active parameter list should be empty at the start of such an iteration.

The final behavioural description (Figure 1: Stage 3) will be determined by the values of key parameters in each component and in each boundary's transfer lists.

6. Conclusions

In this paper a conceptual architecture for modelling abstract-physical systems has been outlined. In particular qualitative causal networks have been introduced as a tool for modelling physical phenomena.
It has been demonstrated how this network structure facilitates the presentation of a localised perspective of the whole system to each component. Boundary links are used to present summary effects to a component allowing qualitative simulation to be done without reference to the source or destination of side-effects. This is an important division of work from the design point of view Causal network management and qualitative simulation can be developed as self-contained, independent modules.

A role for qualitative simulation in modelling physical phenomena has been established. The importance of deep knowledge for 'intelligent reasoning systems has been emphasised in this architecture. The need for shallow rule-based inference systems is also recognised and incorporated into the proposed system to initialise the domains for qualitative simulation and finally in model optimisation

Having outlined the overall modelling process, the prototype implementation will begin with the primary simplification and concentrate on the derivation of a behavioural description in the manner outlined in Section 5.

Acknowledgements

This work is sponsored by the Hitachi Dublin Laboratory.We would like to thank Nobotushi Sagawa and Neil Hurley for various discussions on the subject and comments on earlier drafts of this paper.

References

[Baehman et al 88] P. L. Baehmann, M. S. Shephard, R. A. Ashley and A. Jay. 1988. "Automated metal forming modelling utilizing adaptive re-meshing geometry". SCOREC Report #7 - 1988, Scientific Computation Research Centre, Rensselear Polytechnic Institute.

[Baehman 88] P. L. Baehmann. 1988. "Automated metal forming modelling utilizing adaptive re meshing geometry" SCOREC Report

#9 - 1988, Scientific Computation Research Centre, Rensselear Polytechnic Institute.

[Bobrow '84] D. G. Bobrow. 1984. Qualitative reasoning about physical systems: An introduction. *In Artificial Intelligence,* 24 (1984) 1-5.

[Collar '90] R. R. Collar. 1990. "Automatic idealization control for geometric simplifications in two-dimensional stress analysis". SCOREC Report #13 - 1990, Scientific Computation Research Center, Rensselear Polytechnic Institute.

[de Kleer and Brown '83] J. de Kleer and J. S. Brown. 1983. "The origin, form and logic of qualitative physical laws". In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* 1158 - 1169.

[de Kleer and Brown '84] J. de Kleer and J. S. Brown. 1984. "A qualitative physics based on confluences". In *Artificial Intelligence,* 24, 1984, 7-83.

[Forbus '84] K.D. Forbus. 1984. "Qualitative Process Theory". In *Artificial Intelligence,* 24, 1984, 85-168.

[Forbus et al '87] K.D. Forbus, Paul Nielsen, and Boi Faltings. "Qualitative Kinematics: a framework". In *Proceedings of 1987 International Joint Conference on Artificial Intelligence,* 430-435.

[Gaffney et al '83] P.W. Gaffney, J. W. Wooten, K. A. Kessel, and W. R. McKinney. 1983. "NITPACK: An interactive tree package". *ACM Trans, on Mathematical Software,* Vol. 9, No. 4, December 1983, Pages 395-417

[Gaffney '86] P. W. Gaffeny et al. NEXUS: "Towards a problem solving environment (PSE) for scientific computing", *ACM-SIGNUM Newsletter,* 21:3, July 1986, 13-2.

[Gelsey '90] A. Gelsey. 1990. "Automated physical modelling". In *Proceedings of 1989 International Joint Conference on Artificial Intelligence,* 1225-1230.

[Kuipers '82] B. Kuipers. 1982. "Getting the envisionment right". In *Proceedings of the American Association of Artificial Intelligence,* 209-210.

[Kuipers '84] B. Kuipers. 1984. "Commonsense reasoning about causality: Deriving behaviour from structure". In *Artificial Intelligence,* 24 (1984) 169-203.

[Kuipers '85] B. Kuipers. 1985. "The limits of qualitative simulation". In *Proceedings of the International Joint Conference on Artificial Intelligence,* 128-136.

[Kuipers '86] B. Kuipers. 1986. "Qualitative Simulation". In *Artificial Intelligence,* 29 (1986). 289-338.

[Nayak et al '90] P. P. Nayak, S Addanki, and Leo Joskowicz. 1990. "Modelling with context dependent behaviours". To be published 1991.

[Shephard '88] M. S. Shephard. 1988. "The specification of physical attribute information for engineering analysis". In *Engineering with Computers,* 4, 145-155 (1988).

SOLIDIFICATION OF VARIABLE PROPERTY MELTS IN CLOSED CONTAINERS: MAGNETIC FIELD EFFECTS

George S. Dulikravich
Associate Professor
Aerospace Eng. Dep.
Penn State University
University Park, PA

Branko Kosovic
Graduate Student
Aerospace Eng. Dep.
Penn State University
University Park, PA

Seungsoo Lee
Research Scientist
Aerodynamics Department
Agency for Defense Development
Daejon, South Korea

Abstract: A computer code has been developed for the numerical prediction of steady, laminar, incompressible flows with strong heat conduction, magnetic field effects (Lorentz forces and Joule heating), latent heat of phase change, and thermal buoyancy using extended Boussinesq approximation. The same code predicts the fluid flow field and the solid layer resulting from strong wall cooling. Numerical results for solidification inside a closed container demonstrate the influence of strong magnetic fields on the melt flow field and the solid/liquid interface geometry.

I. INTRODUCTION

Based on our earlier works [1-3] in computational magnetohydrodynamics (MHD) for steady, laminar, incompressible flows in two and three dimensions, we have recently developed a computer code that is capable of simultaneously predicting details of the melt flow field and the formation of the solidified region [4].

Boussinesq approximation was used to account for the thermal buoyancy force, while allowing the coeficients of viscosity, heat conduction, and specific heat to depend on temperature arbitrarily [5]. Nevertheless, in the present work, values of these coefficients were kept constant within the liquid, allowed to vary linearly between liquidus and solidus temperatures, and then again kept constant within the solid. A special test run (b) was performed where the coefficient of viscosity was varied according to the arctangent law over the entire range of temperatures. This computing logic enables us to use a single flow field analysis code in order to simultaneously predict both the fluid flow field and the temperature field inside the accruing solid, thus "capturing" the solid/liquid interface shape without any special front tracking algorithm.

II. ANALYTICAL MODEL

Navier-Stokes equations for incompressible electrically conducting homo-compositional fluid flow are given by

$$v_{i,i} = 0 \qquad (1)$$

$$v_{i,t} + \left(v_i v_j - \frac{Ht^2}{RmRe} H_i H_j\right)_{,j} = -p^*_{,i} + \frac{1}{Re}(\eta' v_{i,j})_{,j} - \frac{Gr}{Re^2} e_i \theta \qquad (2)$$

where p* is the combination of the hydrostatic, hydrodynamic, and magnetic field pressure. Physical properties can be general functions of temperature, while an equivalent specific heat c_{pe}' incorporates the latent heat. Then, energy conservation equation becomes

$$\theta_{,t} + \left(v_j \theta\right)_{,j} = \frac{1}{PrRec_{pe}'}(k'\theta_{,j})_{,j} + \frac{EcHt^2}{Rm^2Rec_{pe}'} \varepsilon_{ijk}\varepsilon_{ilm} H_{k,j} H_{m,l} \qquad (3)$$

while magnetic transport equation is

$$H_{i,t} - \left(v_j H_i - v_i H_j\right)_{,j} = \frac{1}{Rm} H_{i,jj} \qquad (4)$$

The system of governing partial differential equations (1-4) was iteratively integrated using explicit Runge-Kutta four-stage time stepping, and an artificial compressibility formulation [1], except that the velocity components and their derivatives were explicitly set to zero at every point where the instantaneous temperature is lower than the solidus temperature.

III. RESULTS

In a conventional case of a thermal buoyancy induced flow inside a closed rectangular container, the bottom wall is uniformly hot and the top wall is uniformly cold, while the vertical walls were thermally insulated. The computed velocity vector field [2] for such a test case is depicted in Figure 1a. If a uniform vertical downward-pointing magnetic field is added (but without allowing for solidification), the number of recirculating flow regions will change (Fig. 1b) indicating strong influence of the magnetic field on the flow pattern, that is, the reduction of vorticity.

To demonstrate the capability of the code to predict the formation of the solid region, the top of the container was uniformly undercooled. The solid phase was predicted to grow from the top wall (Fig.2). Figure 2 shows the velocity vector fields for the cases with. Ht = 0 (Fig. 2a), Ht = 0 with variable viscosity (Fig 2b), Ht = 5 (Fig. 2c), and Ht = 10 (Fig. 2d). It can be seen that with the increase in the strength of the magnetic field, the vorticity diminishes and the recirculation cell patterns change. Isotherms for the same sequence of test runs are shown in Figure 4. The convergence histories for this test case are represented in terms of an instantaneous count of the solidified cells (Fig. 3) The convergence is oscillatory with a clear indication that the change in the character of the cell from a liquid to a solid (or vice versa) will locally add (or consume) a large amount of energy in the form of latent heat, thus temporarily disturbing the iterative process of simultaneously satisfying all equations in the system. It should be pointed out that the computational grid is fixed and it is clustered towards the container walls. Consequently, the grid is coarse in the central region of the container where the actual solidification occurs. A simple remedy could be a solution-adaptive grid, that is, a grid that is continuously adjusted during the iteration process so that it conforms in a highly clustered pattern with the solid/liquid interface. An even simpler approach could be to increase the number of grid cells through the entire computational domain, thus covering even the unknown interface region with a relatively fine grid. Figure 2 shows clearly the supression of the vorticity due to the increase in the magnetic field strength. Isotherms for the same sequence of runs (Fig. 4) indicate that the fluid/solid interface becomes much smoother with the increase of the magnetic field strength.

IV. REFERENCES

1. Lee, S. and Dulikravich, G S ," Magnetohydrodynamic Flow Computations in Three Dimensions", AIAA Paper 91-0388, Aerospace Sciences Meeting, Reno, NV, January 7-10, 1991;to appear in Int. J. Num. Meth. in Fluids, 1991.
2. Lee, S., Dulikravich, G.S. and Kosovic, B.,"Interaction of Magnetic Field With Blood Flow", Proceedings of the 17th Annual Northeast Bioengineering Conf., Univ. of Connecticut, Hartford, CT, April 4-5, 1991.
3. Lee, S. and Dulikravich, G.S.,"Computation of Magnetohydrodynamic Flows With Joule Heating and Buoyancy", Proceedings of the International Aerospace Congress, Melbourne, Australia, May 12-16, 1991.
4. Dulikravich, G.S., Kosovic, B. and Lee, S.,"Solidification in Channel Flows With Magnetic Fields and Temperature Dependent Physical Properties", submitted for presentation at the ASME WAM, Atlanta, GA, Dec. 1 6, 1991.
5. Gray, D.D. and Giorgini, A., "The Validity of the Bousinesque Approximation for Liquids and Gases", Int. J of Heat and Mass Transfer, Vol. 19, pp. 545-551, 1976.

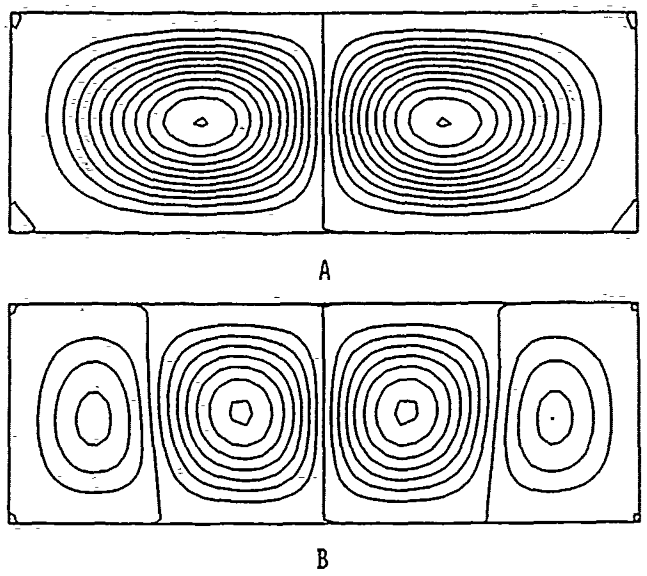


Figure 1. Recirculation streamlines inside the closed container without solidification and with Pr = 7.9, Gr = 3000, Re = sqrt(Gr), Ec = 1 for: a) Ht = 0; b) Ht = 5.

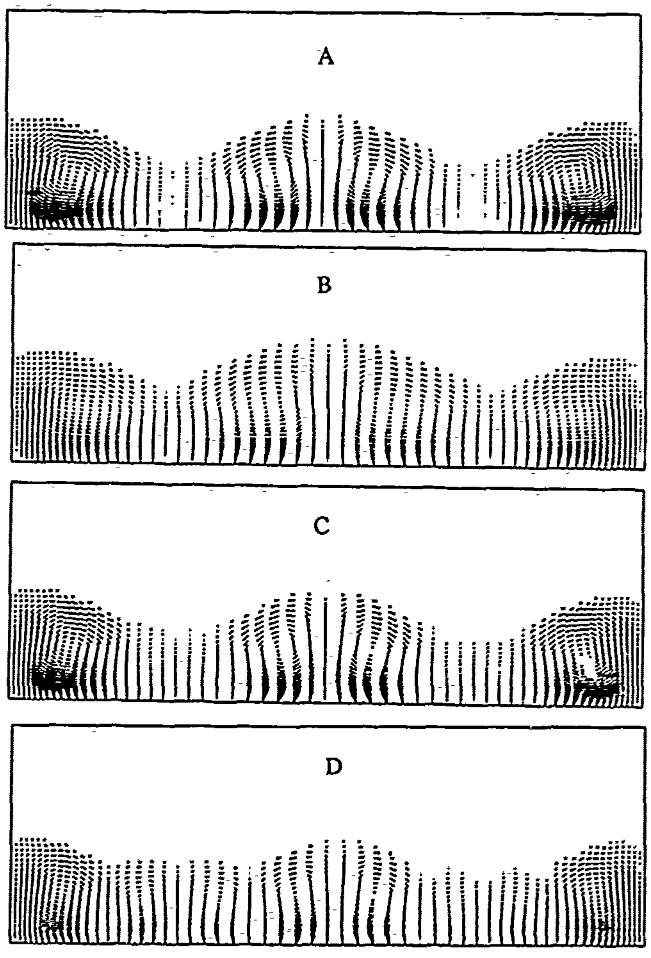


Figure 2. Velocity vector field inside the melt with Pm = 1, Pr = 7.9, Gr = 3000, Ec = 1, Re = sqrt(Gr) and: a) Ht = 0; b) Ht = 0 and variable viscosity; c) Ht = 5; d) Ht = 10.

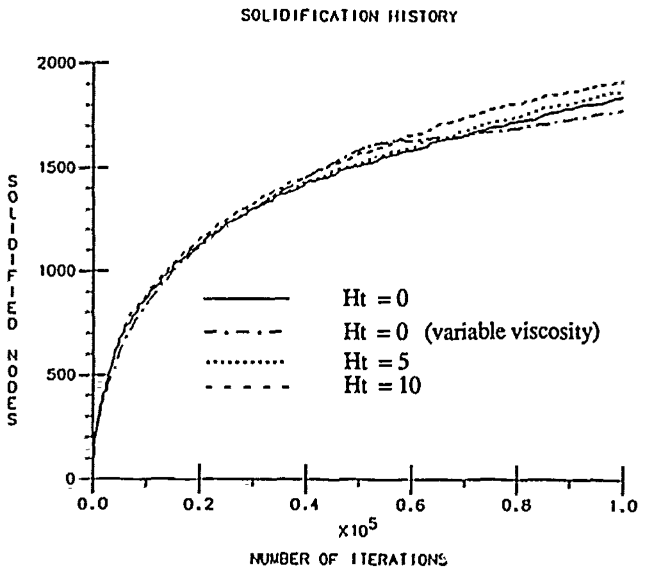


Figure 3. Convergence histories: number of solidified cells versus number of iterations.

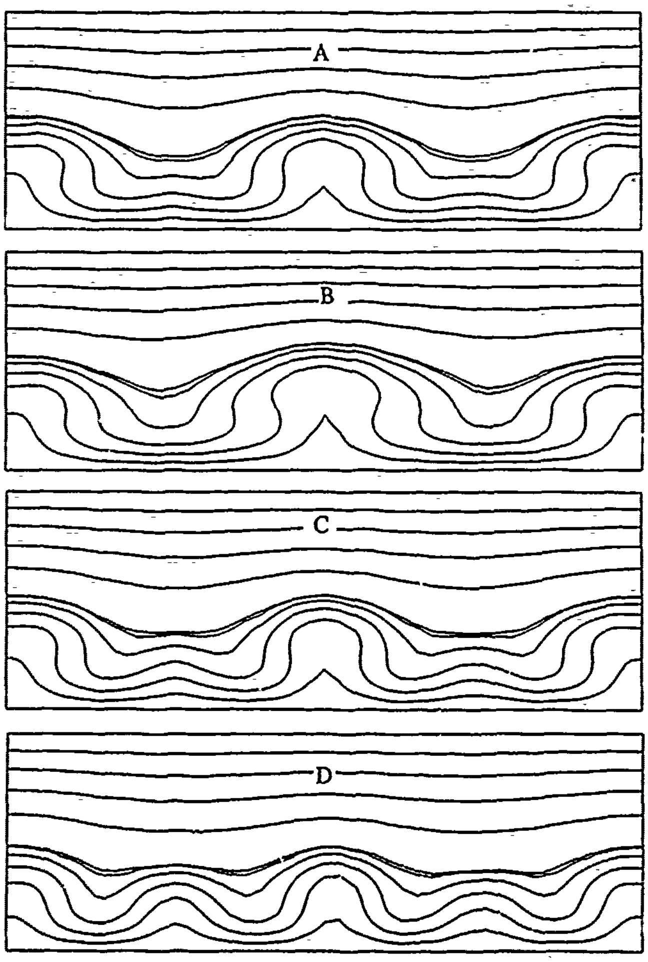


Figure 4. Isotherms inside the solid and the melt with Pm = 1, Pr = 7.9, Gr = 3000, Ec = 1, Re = sqrt(Gr) and: a) Ht = 0; b) Ht = 0 and variable viscosity; c) Ht = 5; d) Ht = 10.

AN ASYMPTOTIC MODEL FOR DETONATION IN DUCT

S. GERBI

ENS Lyon,
46 allée d'Italie, 69364 LYON CEDEX 07
France

G. CLAUS

ELF - CRES,
Chemin du Canal, BP 22, F
69360 ST SYMPHORIEN D'OZON, FRANCE

Abstract- An asymptotic model for a detonation in duct is derived through activation energy asymptotics. Using linearized stability analysis, we study the stability of the constant equilibrium state ; next we investigate the quenching phenomenon. Finally we study a bifurcation phenomenon related to the length of the duct and propose a simplified model.

1- Introduction

The steady detonation structure consists of a shock wave, followed by an induction zone in which reaction is weak, followed by a zone in which vigorous reaction and heat release occurs. Behind the reaction zone is uniform burnt gas. In the limit of large activation energy ($\theta \rightarrow \infty$) this structure reduces to the well known square-wave in which conditions are uniform in the induction zone (of length L) and the reaction zone is a discontinuity.

When the stability of such detonation wave is examined, for a certain class of disturbances based upon L and θ, [1], [2] perturbations to the shock displacement are governed by the fully nonlinear parabolic equation [3]

$$(1) \qquad g_t + \frac{1}{2}\, g_x^2 = \ln \left(\frac{e^{cgg_{xx} - 1}}{cg_{xx}}\right)$$

associated with the natural Neumann boundary conditions at the walls :

$$(2) \qquad g_x\,(0,t) = g_x\,(l\,,t) = 0$$

in which c is a positive given constant and l is the width of the duct. In this presentation, we are interested in the stability of steady-state solutions of problem (1)-(2). Of course $g \equiv 1$ is the expected unperturbed solution but numerical investigation has shown the existence of periodic non-constant solutions for a certain range of parameter l. Two kinds of results have been obtained for the evolution of a perturbation of the constant solution $g \equiv 1$.

2. Linearized stability analysis

Linearizing problem (1)-(2) around $g \equiv 1$, we get the dynamical system (3)-(4)

$$(3) \qquad u_t = \mathcal{L} u = \frac{c}{2} u_{xx} + u$$

$$(4) \qquad u_x\,(0,t) = u_x(l\,,t) = 0:$$

The spectrum of $\mathcal{L}$ consists of the eigenvalues $\lambda_j = 1 - j\,\dfrac{c\pi^2}{2l^2}$ with corresponding eigenvectors $\omega_j = \cos j\,\dfrac{\pi x}{l}$, $j = 0, 1, \ldots$

There is at least one unstable mode (the planar mode) corresponding to $\lambda_0 = 1$. We define the critical length $l_c = \pi \sqrt{\frac{c}{2}}$ for which $\lambda_1 = 0$, leading to a center manifold. For $l < l_c$, the stable manifold is codimension 1. Numerically, we validate this behaviour on the nonlinear problem (1)-(2) by taking a small initial condition with null mean-value : the evolution of such a perturbation vanishes [4].

3. Quenching phenomenon

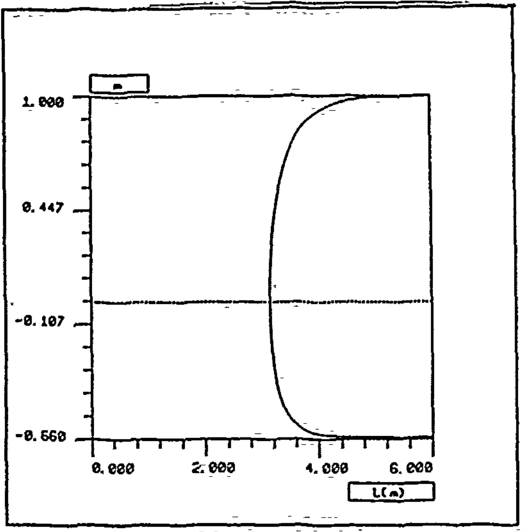
This result is also related to the previous stability analysis : taking a non periodic initial condition leads to an unstable behavior. Nevertheless, the solution remains bounded whereas the time derivative g_t blows-up in finite time t_c : this is the so-called quenching phenomenon.

More precisely, an analytical argument [4] shows that : $g_{\sim}(t - t_c) \log (t - t_c)$

4. Critical length l c and bifurcation phenomenon

As previously mentionned, there exists a critical length l_c for which λ_1 crosses 0. Therefore as l increases, crossing l_c, the dimension of the stable manifold decreases. On the other hand, we exhibited nontrivial solutions for $l > l_c$. Figure 1 gives a description of the steady-state periodic solutions in the plane

$(\sqrt{\frac{2}{c}}\, l\,, g(0)\,)$.



(Figure 1)

This is the typical exchange of stability occuring when there exists a branch of bifurcation. In order to give a mathematical description of this bifurcation phenomenon, we derive a simpler model whose behavior can be compared to the physical one. Assuming $|g|$ and $|g_{xx}|$ to be small and taking a first order approximation of the logarithmic term we get :

(5) $$u_t - u_{xx} = \frac{-2l^2}{c}(1 - u)\log(1 - u)$$

(6) $$u_x(0,t) = u_x(0,1) = 0$$

where u is defined by

(7) $$g = 1 - c\log(1 - u)$$

On this model, we study the existence of non-trivial stationary solutions and their stability as well as the quenching phenomenon.

References:

[1] J. D. Buckmaster, A theory for triple point spacing in overdriven detonation waves, Combustion & Flame 77, (1989), 219-288.

[2] C. Schmidt-Lainé, W. Dold, J. Buckmaster, Pressure spot formation in unstable detonation waves. Proc. 3rd. Int. Num. Combustion. Sophia-Antipolis. B. Larrouturou Ed. Springer-Verlag.

[3] G. Da Prato, A. Lunardi, Stability, instability and center manifold theorem for fully nonlinear parabolic equations in Banach space, Arch. Rat. Mech. Anal 101 (1988) p 115-141.

[4] C. M. Brauner, J. D. Buckmaster, J. W. Dold, C. Schmidt-Lainé: On an evolution equation arising in detonation theory, to appear in Fluid Dynamical Aspects of Combustion theory. M. Onofri and A. Tesei eds. Longman Publ.

Homogenization for nonlinear adsorption-diffusion processes in porous beds

Éric Canon

Université de Saint-Étienne

23 rue du docteur Paul Michelon

F-42023 Saint-Étienne Cédex 2.

Abstract : A homogenized model for nonlinear adsorption diffusion processes in porous beds, modelling chromatographic columns, is proposed. We consider an inhomogeneous periodic medium with two levels of structure having strong different scales. This suggests to homogenized it. The first structure is a convection area.The second structure is an adsorption area, made of small porous cristals. Our equations are convection-diffusion equations for concentration, and Stokes equations for the fluid velocity. Along the cristal boundary, we have three kinds of discontinuity : jumps of the fluid velocity and of the diffusion coefficient, and a nonlinear jump for the concentration. This last heterogenity is the most important in view of homogenization. We show a maximum principle ans some energy estimates for our nonlinear problem. From them, we prove, as main result, the convergence of our microscopic model to a homogeneous non linear macroscopic model.

I. INTRODUCTION

This paper presents a general homogenized macroscopic model for convection-diffusion equations into a chromatographic column. This model takes into account the complete internal heterogeneous structure of the column, and nonlinear relationships between moving and adsorbed phases. The main point of our problem is this nonlinearity. this discontinuity is an isothermal relationship derived from termodynamical considerations. A chromatographic column is constituted, at a microscopic scale, of two structures having strong different scales: a convection area and an adsorption area made of small porous cristals. These structures are assumed to be periodic. Our equations are convection-diffusion equations coupled with Stokes equations. The difference of scales between the two levels suggests us to use the homogenization techniques to replace the heterogeneous medium by an equivalent homogeneous medium.

The structures are assumed to depend on some small parameter ε. This parameter will tend to zero in the homogenization process. We shall show that under some reasonable assumptions on the isotermal jump condition, the model converges to some nonlinear integrodifferential equation. The ε–depending model will be called the microscopic model. The homogeneous limit model willbe the macroscopic model.

II. THE MICROSCOPIC MODEL

Our equations are convection-diffusion equations coupled with Stokes equations. The heterogeneities are: jumps of discontinuity for the velocity and for the diffusion coefficient, and a nonlinear jump of discontinuity for the concentration The last one is the most important. Our domain Ω is divided into two parts Ω_1^ε and Ω_2^ε, corresponding to the inhomogeneous parts of the column. Γ^ε denotes the boundary between Ω_1^ε, Ω_2^ε. The unknown function is the corresponding concentration denoted by w_i^ε, i=1, 2. The fluid velocity is denoted by u^ε, and the coefficient of diffusion by D_i, i=1, 2. The geometry of Ω_1^ε, Ω_2^ε, Γ^ε, and consequently the solutions w_i^ε, and u^ε depend on ε. We have to homogenize the following microscopic model :

--> Convection-diffusion in Ω_1^ε :

$$\partial_t w_1^\varepsilon = D_1 \Delta w_1^\varepsilon - u^\varepsilon . \nabla w_1^\varepsilon \qquad (1)$$

--> Convection in Ω_2^ε:

$$\partial_t w_2^\varepsilon = \varepsilon^{2} D_2 \Delta w_2^\varepsilon \qquad (2)$$

--> Jumps of discontinuity for the concentration ·

$$w_2^\varepsilon = h(w_1^\varepsilon) \quad \text{on } \Gamma^\varepsilon \qquad (3)$$

--> Flux continuity on Γ^ε:

$$D_1 \nabla w_1^\varepsilon . n = \varepsilon^{2} D_2 \nabla w_2^\varepsilon . n \qquad (4)$$

--> Dirichlet conditions on the input part of the boundary .

$$w_1^\varepsilon = v_0 \quad \text{on } \Gamma_{in} \qquad (5)$$

--> Neumann condition on the impermeable wall and on the output part of the boundary :

$$\nabla w_1^\varepsilon . n = 0 \quad \text{on } \Gamma_0 \cup \Gamma_{out} \qquad (6)$$

--> Initial condition :

$$w^\varepsilon (x, t = 0) = 0. \qquad (7)$$

--> Stationary Stokes equations for the fluid velocity in Ω_1^ε :

$$\Delta u^\varepsilon = \nabla p^\varepsilon \qquad (8)$$

$$\text{div } u^\varepsilon = 0 \qquad (9)$$

with the following boundary conditions:

$$u^\varepsilon = 0 \quad \text{on } \Gamma^\varepsilon \cup \Gamma^0 \ \text{(and in } \Omega_2^\varepsilon) \qquad (10)$$

$$u^\varepsilon = u_0 \quad \text{on } \Gamma^{in} \cup \Gamma^{out} \qquad (11)$$

$$u^\varepsilon . n \leq 0 \quad \text{on } \Gamma^{in} \qquad (12)$$

$$\int_{\Gamma^{out}} u^\varepsilon . n = - \int_{\Gamma^{in}} u^\varepsilon . n \qquad (13)$$

III. THE MACROSCOPIC MODEL

In this section we give the macroscopic model which is defined to be the limit model of the microscopic model stated in the previous section when, the parameter ε goes to zero the convergence will be examinated in the next section. Notice that this model is uniform for the whole domain Ω. The macroscopic model is defined as follow (where $\overline{w}$ denote the concentration in Ω and u the fluid velocity) :

$$\frac{|Y^1|}{|Y|} \partial_t \overline{w} + f(t) * \partial_s h(\overline{w}) - \sum_{i,j=1}^{3} a_{ij} \partial_i \partial_j \overline{w} + u . \nabla \overline{w} = 0 \qquad (14)$$

The deformation of the Laplace operator due to the geometry of the heterogeneity is given by :

$$a_{ij} = D_1 \frac{|Y^1|}{|Y|} [\delta_{ij} + (\frac{1}{|Y^1|} \int_{Y^1} \nabla \sigma_k(y) \, dy) \qquad (15)$$

where : σ_k is a Y-periodic function in $H^1 (Y_1)$ (Y is a basic cell, Y_1 is the convection part of the cell Y) defined as :

$$\Delta \sigma_k = 0 \text{ in } Y_1 \qquad (16)$$

$$\nabla\sigma_k.n = -n_k \text{ on } \partial Y_1 \tag{17}$$

The convolution term f expressing the adsorbed phase is given by:

$$f(t) = \frac{1}{|Y|}\frac{d}{dt}\int_{Y_2} \rho_0(t,y)\,dy \tag{18}$$

where ρ_0 is a Y-periodic function in H^1 (Y_2) (Y_2 is the adsorption part of the cell Y) defined as :

$$\partial_t \rho_0(y,t) - D_2\,\Delta\,\rho_0(y,t) = 0 \text{ in } Y_2 \tag{19}$$

$$\rho_0(y,t) = 1 \text{ on } Y_1 \cup \partial Y_2 \tag{20}$$

$$\rho_0(y,0) = 0 \tag{21}$$

The fact that the adsorbed phase appears in (14) as a memory term is strongly related to the fact that equation (2) in the microscopic model is linear. For a more general model, we get a coupled system in place of equation (14) : a convective-diffusive equation for the whole domain Ω with a source term for the adsorption process, coupled with a diffusive cell equation in Y_2 involving h on the boundary ∂Y_2.

IV. CONVERGENCE

It is wellknown (Tartar) that the homogenization of the Stokes's equations in porous media leads to the Darcy's law. We do not rewrite it here.

We assume the following properties for some extension of the boundary condition v_0 in (5) :

$$v_0 \in L^2(0,T;\, H^1(\Omega)) \cap H^1(0,T;\, L^2(\Omega)) \cap L^\infty([0,T] \times \Omega) \tag{22}$$

$$\partial_t v_0 \in L^2(0,T;\, H^1(\Omega)) \cap H^1(0,T;\, L^2(\Omega)) \cap L^\infty([0,T] \times \Omega) \tag{23}$$

The convergence of the microscopic model to the macroscopic model is obtained for the weak formulation of the models given in the two previous sections. It follows of the a priori estimates stated in the two following lemmas:

Lemma 1 (maximum principle) :

Under hypothesis (22) and (23) the following estimates hold for the solution w^ε of the microscopic problem :

$0 \leq w^\varepsilon(x,t) \leq Sup\,(v_0(x,t)\,;\,(x,t) \text{ in } \Gamma_{in} x [0,T])$ for almost every (x,t) in $\Omega \times [0,T]$.

$0 \leq \partial_t w^\varepsilon(x,t) \leq Sup\,(\partial_t v_0(x,t)\,;\,(x,t) \text{ in } \Gamma_{in} x [0,T])$ for almost every (x,t) in $\Omega \times [0,T]$.

The prove is based on Stampacchia's method.

Lemma 2 (energy estimates) :

For the solution w^ε of the microscopic problem the following norms are bounded independently of ε :

$$\|w_1^\varepsilon\|_{L^\infty(o,T;L^2(\Omega_1^\varepsilon))},\ \|w_2^\varepsilon\|_{L^\infty(o,T;L^2(\Omega_2^\varepsilon))},\ \|\nabla w_1^\varepsilon\|_{L^2(o,T;L^2(\Omega_1^\varepsilon))},$$
$$\varepsilon\|\nabla w_2^\varepsilon\|_{L^2(o,T;L^2(\Omega_2^\varepsilon))},\ \|\partial_t w_1^\varepsilon\|_{L^\infty(o,T;L^2(\Omega_1^\varepsilon))},\|\partial_t w_2^\varepsilon\|_{L^\infty(o,T;L^2(\Omega_2^\varepsilon))},$$
$$\|\partial_t \nabla w_1^\varepsilon\|_{L^2(o,T;L^2(\Omega_1^\varepsilon))},\ \varepsilon\|\partial_t \nabla w_2^\varepsilon\|_{L^2(o,T;L^2(\Omega_2^\varepsilon))}.$$

Let us define $\overline{w}^\varepsilon$ the h-harmonic extension of w^ε :

$$\Delta\,h\,(\overline{w^\varepsilon}) = 0 \text{ in } \Omega_2^\varepsilon. \tag{24}$$

We have the following convergence result :

Theorem :

There exists a unique function $\overline{w}$ of H^1 $(0,T\,;\,V)$ which is the limit of the family $\overline{w}^\varepsilon$ (solution of the weak microscopic problem) in H^1 $(0,T;V)$. This limit function satisfies the weak formulation of the nonlinear integrodifferential equation (14).

Acknoledgments : this work was supported by the Societé Nationale Elf Aquitaine (SNEA).

References :

Canon É., Jäger W. : *A homogeneized model for nonlinear diffusive chromatography.* 1991, to appear.

Sanchez-Palencia E. : *Non-homogeneous media and vibration theory.* Lecture Notes in Physics, No 127. Springer Verlag 1980

Vogt C. : *A homogenization theorem leading to a Volterra-integrodifferential equation for permeation chromatography.* Universität Heidelberg, 1982, N0 155.

A PHASE FIELD MODEL FOR ISOTHERMAL SOLUTION GROWTH

A. A. WHEELER
National Institute of Standards and Technology
Gaithersburg, MD 20899 U.S.A.

and

W. J. BOETTINGER
National Institute of Standards and Technology
Gaithersburg, MD 20899 U.S.A.

ABSTRACT – We describe a new phase field model for the phase transition of an isothermal binary alloy. This is the first time, to the authors knowledge, that a phase field model has been proposed for phase transition in a impure material. This represents a significant step in the derivation of a phase field model for the solidification of a nonisothermal binary alloy.

I. INTRODUCTION

Classical macroscopic models of phase transitions model the interface between regions of different phase as a surface, and hence assume it has zero thickness. The governing equations for thermodynamic variables, such as temperature and solute, are formulated in each phase independently, based upon conservation principles and quantitively verified phenomenological laws. The boundary conditions at the interface are chosen to describe the processes, such as liberation of latent heat and segregation that occur at the interface on a microscopic scale. This approach gives rise to the formulation of a free boundary problem which provides a difficult mathematical setting and only the simplest models of phase change have been rigorously mathematically analysed. The advantage of these models is that it is clear from the outset what physical mechanisms are incorporated into them, and comparison with careful controlled experiments is possible.

An alternative technique for investigating transport processes in systems involving a phase transition, involves the construction of a Landau-Ginzberg free energy functional. This approach has its roots in statistical physics, Landau and Khalatinikov, [1]. Further, a phase field, which is a function, $\phi(x,t)$ is postulated which describes the phase of the system at any point in time and space. It is assumed that the Helmholtz free energy $\mathcal{G}(\phi, \quad)$, is a functional of the phase field, as well as any other thermodynamic variables, (such at temperature which are denoted here by ellipsis) in the following way:

$$\mathcal{G}(\phi,\ldots) = \int_{\Omega} \tfrac{1}{2}\epsilon^2(\nabla\phi)^2 + g(\phi,\ldots)d\Omega, \qquad (1)$$

where Ω is the region occupied by the system, and $g(\phi,\ldots)$ is the free energy density. Its dependence on ϕ usually has a "double well" form. The phase field is then assumed to evolve as:

$$\dot{\phi} \propto L\left(\frac{\delta\mathcal{G}}{\delta\phi}\right), \qquad (2)$$

where L is some partial differential operator. This equation is then supplemented by partial differential equations for the other thermodynamic variables. Cahn, [2], has successfully used this approach to model spinodal decomposition of a binary alloy, although here the concentration naturally plays the role of the phase field. Various models that employ this idea are reviewed by Halperin, Hohenburg and Ma, [4], particularly in regard to the study of critical phenom ena. The Model C given by these authors has been adapted by Langer, [5], Fix, [6] and most prolifically by Caginalp, [7] to derive the so-called "phase-field model" of solidification which models the phase change of a pure material. Caginalp has extensively studied this, and variations of this model, [8] [9]. It has emerged from study of this model that qualititively it exhibits features common to solidification of a pure material. Numerical calculations based on this model, by Smith, [10] and a similar model, by Kobayashi, [11], show breakdown of a planar and circular interfaces to cellular structures, as well as the formation of dendrite like structures, liquid trapping and coarsening behaviour.

Caginalp, [8] has shown in various distinguished limits, in which $\epsilon \to 0$, that various forms of the classical Stefan problem may be recovered, in which the interface is taken to be "sharp" i.e. modeled by a surface. In this limit there are thin layers within Ω of thickness $O(\epsilon)$ in which the phase field rapidly changes. These are interpreted as representing interfaces, which are necessarily diffuse. From this analysis it transpires that in some limits, the interfacial dynamics involve curvature effects corresponding to the Gibbs-Thompson interfacial surface energy as well as kinetic effects. Further, it is also possible to recover the classical Hele-Shaw problem in other limits. It is clear that this approach can embody a considerable variety of realistic physical effects in a coherent way.

However, this superabundance of physical phenomena also provides a difficulty when applying the model to a definite physical situation. This is because it is not clear how to choose the values of parameters in the phase field model so that it models the solidifica tion of a pure material with given materials and growth parameters, (or equivalently the Stefan number and capillary number).

Another difficulty with this particular model, as pointed out by Penrose and Fife [12], is that it is thermodynamically inconsistent. This is because the free energy functional is *only* employed in the formulation of the kinetic equation for the phase field. The concern here is that the solution of the above governing equations does not correspond to the free energy decreasing monotonically with time, as required by the Second Law of Thermodynamics. An alternative approach suggested by these authors is to construct an entropy functional, $\mathcal{S}$, of the system and require it to evolve as a gradient flow of the form:

$$\dot{u} \propto \mathrm{grad}_0\mathcal{S}(u), \qquad (3)$$

where grad_0 is a suitable constrained gradient, and u represents the thermodynamic variables. This formulation necessarily ensures that the total entropy of the system increases with time.

The appeal of phase field models in describing phase transitions is twofold;

- It provides a simple, elegant description, that appears to em body a rich variety of realistic physical phenomena.
- From a computational point of view it is relatively simple to compute solutions. This is because it is not necessary to distinguish between the different phases. Computations on the classical sharp interface formulation require that the free boundary is tracked numerically and that the region occupied by each phase is therefore determined and dealt with individually. This results in very difficult and untidy numerical algorithms.

In this paper we derive a new phase field model for phase transitions of an isothermal binary solution. To our knowledge, there are to date no phase field models that deal with impure materials. The model presented here is a first step to developing a phase field model for the solidification of an alloy.

II. PHASE FIELD MODEL

We consider an isothermal solution of two different species A and B in which are present two phases, solid and liquid, contained in a fixed region Ω with boundary $\partial\Omega$. We denote the concentration of B by $c(x,t)$ and we introduce a phase field $\phi(x,t)$ which represents the phase in time and space in Ω. For definiteness we describe the solid liquid interface by $\phi(\mathbf{x},t) = \frac{1}{2}$ and denote solid regions where $\phi(\mathbf{x},t) > \frac{1}{2}$ and liquid regions where $\phi(\mathbf{x},t) < \frac{1}{2}$.

A recent phase field model due to Kobayashi [11] models the phase transition of a pure material by employing the following gradient weighted free-energy functional:

$$\mathcal{F}_K(\phi,T) = \int_\Omega \frac{\epsilon^2}{2}|\nabla\phi|^2 + f_K(\phi,T)d\Omega, \tag{4}$$

where ϵ is a constant, $T(\mathbf{x},t)$ is the temperature and the free energy density $f(\phi,T)$ is represented by:

$$f_K(\phi,T) = \int^{\phi} p(p-1)(p-\frac{1}{2}-\beta(T))dp, \tag{5}$$

where $\beta(T)$ is a monotonic increasing function of T, such that $\beta(T_M) = 0$, where T_M is the freezing temperature of the material and $|\beta(T)| < \frac{1}{2}$. The free energy density $f_K(\phi,T)$ is a double well potential. The restriction $|\beta(T)| < \frac{1}{2}$ ensures that it has local minima at $\phi = 0$ and $\phi = 1$, and a local maxima at $\phi = \frac{1}{2} + \beta(T)$. Because of the two minima the system may exist stablely in a state which is all-liquid ($\phi(\mathbf{x},t) = 0$) or all-solid ($\phi(\mathbf{x},t) = 1$). There is an energy penalty for a change of phase within the region Ω, which corresponds to ϕ varying between zero and unity. This is because such a variation increases the total energy $\mathcal{F}_K$ of the system, due to an increased energy density associated with the double well nature of the energy density, and also due to the contribution to the total energy due to the gradient energy, which is no longer zero.

If $-\frac{1}{2} < \beta < 0$, then the *global* minima of the energy density is at $\phi = 1$ and so the all-solid state is the one with the lowest energy and is hence the prefered state. However, if $0 < \beta < \frac{1}{2}$ then the situation is reversed and the liquid is the preferred state. We see that at temperatures below the melting point the solid phase has the minimum energy and is prefered, whereas for temperatures above the melting temperature, the all-liquid phase is preferred.

We now employ this form for the free energy density to develop the appropriate free energy density for an isothermal ideal solution. We assume that the temperature, T, which is given, is such that if the solution consisted only of species A ($c \equiv 0$) the all-solid phase would be the prefered state i.e. $T < T_M^A$, where T_M^A is the melting temperature of pure A. Further, we also assume that the temperature T is sufficiently large that if the system consisted of species B alone ($c \equiv 1$) the all-liquid phase would be the prefered state i.e. $T > T_M^B$, where T_M^B is the melting temperature of pure B. We also assume that the *molar* Gibbs free energy densities of each species A and B alone are of the form given by Kobayashi, and are denoted by $f_A(\phi;T)$ and $f_B(\phi;T)$ respectively. Specifically we put.

$$f_A(\phi;T) \;=\; W_A \int^{\phi} p(p-1)(p-\frac{1}{2}-\beta_A(T))dp, \tag{6}$$

$$f_B(\phi;T) \;=\; W_B \int^{\phi} p(p-1)(p-\frac{1}{2}-\beta_B(T))dp, \tag{7}$$

where here W_A, W_B are constants, and T the temperature is a parameter in this isothermal situation. We note that because $T_M^B < T < T_M^A$, then $-\frac{1}{2} < \beta_A(T) < 0 < \beta_B(T) < 1$. The total energy density $f(\phi,c;T)$ of the solution is:

$$f(\phi,c;T) = cf_B(\phi,T) + (1-c)f_A(\phi;T) + \frac{kT}{v_m}[c\log c + (1-c)\log(1-c)], \tag{8}$$

where k is Boltzmans constant and v_m is the molar volume. The first two terms correspond to the contribution to the energy density due to the individual molar Gibbs free energies of the two species and the last term is due to the decrease in energy associated with the mixing of the two constituents, under our assumption that it is an ideal solution.

In a similar way to Kobayashi we define the free energy functional by:

$$\mathcal{F}(\phi,c;T) = \int_\Omega \frac{\epsilon^2}{2}|\nabla\phi|^2 + f(\phi,c;T)d\Omega. \tag{9}$$

In order to derive a kinetic model we make the assumption that the system evolves in time so that its total free-energy decreases monotonically. This may be met by assuming the rate of change of c and ϕ vary according to the constrained gradient of $\mathcal{F}(\phi,c,T)$.

$$\dot{\mathbf{u}} \propto -\mathrm{grad}_0\mathcal{F}(\mathbf{u}), \tag{10}$$

where $\mathbf{u} = (\phi,c)^T$. Fife, [13] discusses how such constrained gradients may defined in a more rigorous mathematical setting. The only constraint we require here is that both species are conserved, i.e. $\frac{d}{dt}\int_\Omega c d\Omega = 0$. We chose the constrained gradient such that:

$$\frac{\partial\phi}{\partial t} = -\kappa_1\frac{\delta\mathcal{F}}{\delta\phi}, \tag{11}$$

$$\frac{\partial c}{\partial t} = \kappa_2\nabla\cdot(c(1-c)\nabla\frac{\delta\mathcal{F}}{\delta c}, \tag{12}$$

where κ_1 and κ_2 are constants. The boundary conditions are

$$\frac{\partial\phi}{\partial n} = \frac{\partial c}{\partial n} = 0, \tag{13}$$

where $\mathbf{n}$ is the outward normal to the boundary $\partial\Omega$. We may interpret the right hand side of (12) as the divergence of a solute flux, $\mathbf{j} = c(1-c)\nabla\frac{\delta\mathcal{F}}{\delta\phi}$. The coefficient $c(1-c)$ has been included to ensure that the diffusion equation for the solute that emerges has a diffusion coefficient that is constant.

Evaluating the variational derivatives of the free energy functional gives that:

$$\frac{\partial\phi}{\partial t} = \kappa_1\left(\epsilon^2\nabla^2\phi - \frac{\partial f}{\partial\phi}\right), \tag{14}$$

$$\frac{\partial c}{\partial t} = \kappa_2\nabla\cdot\left(c(1-c)\nabla\frac{\partial f}{\partial c}\right), \tag{15}$$

which may be also written as:

$$\frac{\partial\phi}{\partial t} = \kappa_1\left[\epsilon^2\nabla^2\phi - \left(c\frac{\partial f_A}{\partial\phi} + (1-c)\frac{\partial f_B}{\partial\phi}\right)\right], \tag{16}$$

$$\frac{\partial c}{\partial t} = \kappa_2\nabla\cdot(c(1-c)\nabla(f_A - f_B)) + D\nabla^2 c, \tag{17}$$

where $D = \frac{\kappa_2 kT}{v_m}$ is the diffusivity of the solute.

References

[1] L. D. Landau and Khalatnikov, I. M., in; Collected works of L. D. Landau, ed. D. ter Haar (Pergamon, Oxford, 1965) pp 626-633.

[2] J. W. Cahn and Hilliard, J. E., J. Chem. Phys. 28 (1958) 258-267.

[3] J. W. Cahn, Acta Metallugica, 9 (1961) 795-801.

[4] B. I. Halperin, Hohenburg, P.C. and Ma, S.-K., Phys. Rev. B. 10 (1974) 139-153.

[5] J. S. Langer, pp 164-186, World Science Publishers (1986)

[6] G. Fix, ed. A. Fasano and M. Primocerio, pp 580-589, Pitman, London (1983).

[7] G. Caginalp, Arch. Rat. Mech. Anal. 92 (1986) 205-245.

[8] G. Caginalp, Phys. Rev. A. 39 (1989) 5887-5896.

[9] G Caginalp and Fife, P. C., Phys. Rev. B. 33 (1986) 7792-7794.

[10] J. Smith, J. Comp. Phys. 39 (1981) 112-127.

[11] Private communication (1990).

[12] O. Penrose and Fife, P. C., Physica D 43 (1990), 44-62.

[13] P. C. Fife, Proc. Taniguchi Int. Symp. on Nonlinear PDEs and Applications, Kinokuniya Pub. Co. (1990).

USE OF SPHERICAL HARMONICS IN THE SOLUTION OF THE RADIATION TRANSFER PROBLEM IN AN ATMOSPHERE WITH THE INHOMOGENEOUS SPREADING SURFACE

SULTANGAZIN U.M.
Institute of Mathematicѕ & Mechanics
Academy of Sciences of Kazakh SSR
Pushkin str.125, Alma-Ata, USSR

AND

MULDASHEV T.Z.
Institute of Mathematics & Mechanics
Academy of Sciences of Kazakh SSR
Pushkin str.125, Alma-Ata, USSR

Abstract-A method is presented for solving the multidimensional equation of radiative transfer in a scattering plane-parallel atmosphere with inhomogeneous spreading surface. The method, based on the spherical narmonics expansion, can be used to compute models with an arbitrary large optical thickness and any scattering phase function.

The correction problem of the distortion of the representation of the earth's surface arises in connection with the investigations of natural resources of the Earth from the space. The main stage here is the numerical solving of the following problem of radiative transfer in three-dimensional plane slab:

$$\begin{cases} (\vec{\omega},\ \mathrm{grad}\ I) + \sigma(z)\ I = S\ I, \\ I|_{z=0} = \pi F_0 \delta(\vec{\omega}-\vec{\omega}_0), \quad \vec{\omega} \in \Omega_+, \\ I|_{z=H} = \frac{q(x,y)}{\pi}\int\limits_{\Omega_+} I\mu'\, d\mu', \vec{\omega} \in \Omega_-, \end{cases} \tag{1}$$

where $S\ I = \frac{\sigma_s(z)}{4\pi}\int\limits_{\Omega} g(z,\mu_s) I(\vec{r},\vec{\omega}')\, d\vec{\omega}'$.

$I(\vec{r},\vec{\omega})$ is the radiation intɔnsity at the point $\vec{r} = (x,y,z)$ along the unit vector $\vec{\omega} = (\sqrt{1-\mu^2}\cos\varphi, \sqrt{1-\mu^2}\sin\varphi, \mu)$, Ω is the surface of the unit sphere, $\Omega_+=\{\vec{\omega} : (\vec{\omega},\vec{n})>0\}$, $\Omega_-=\{\vec{\omega} : (\vec{\omega},\vec{n})<0\}$, $\sigma(z),\sigma_s(z)$ are the extinction and scattering coefficients, respectively, $g(z,\mu_s=\vec{\omega}\cdot\vec{\omega}')$ is the scattering phase function.

Boundary condition at the z=0 defines the illumination of the top of an atmosphere by a unidirectional beam of monochromatic radiation of strength πF_0. Condition at the z=H is the Lambert's reflection law, where $0 \le q(x,y) \le 1$ is the reflection coefficient of the spreading surface.

By the method described I.V.Mishin and T.A.Sushkevich [1] the solution of the problem (1) can be reduced to the solving of the problems (2) and (3).

$$\begin{cases} \mu\frac{\partial\tilde{I}}{\partial z} + \sigma(z)\ \tilde{I} = S\ \tilde{I}, \\ \tilde{I}|_{z=0} = \pi F_0 \delta(\vec{\omega}-\vec{\omega}_0), \quad \vec{\omega} \in \Omega_+, \\ \tilde{I}|_{z=H} = 0, \quad \vec{\omega} \in \Omega_-, \end{cases} \tag{2}$$

$$\begin{cases} \mu\frac{\partial\Psi}{\partial z} + [\sigma-i(p_x\sqrt{1-\mu^2}\cos\varphi+p_y\sqrt{1-\mu^2}\sin\varphi)]\Psi = S\ \Psi, \\ \Psi|_{z=0} = 0,\ \vec{\omega} \in \Omega_+; \quad \Psi|_{z=H} = 1,\ \vec{\omega} \in \Omega_-; \end{cases} \tag{3}$$

The numerical solution of the problem (2) considered in our work [2]. And now we discuss the use of the spherical harmonics method for solving (3) when $p_y=0$.

In this case, $\Psi(z,\mu,\varphi,p_x)$ can be expressed in a Fourier series as $\Psi=\Psi^0+2\sum\limits_{m=1}^{M}\Psi^m\cos m\varphi$. (4)

Phase function allows following expansion:

$$g(z,\mu_s)=\gamma^0(z,\mu,\mu')+2\sum_{m=1}^{\infty}\gamma^m(z,\mu,\mu')\cos m(\varphi-\varphi'), \tag{5}$$

where $\gamma^m = \sum\limits_{k=m}^{\infty} g_k(z)Y_k^m(\mu)Y_k^m(\mu')$, $Y_k^m(\mu)$ is the normalized associated Legendre polynomials.

By substituting (4) and (5) in (3) we obtain

$$\begin{cases} \mu\frac{\partial\Psi^m}{\partial z} + \sigma\Psi^m - ip_x\sqrt{1-\mu^2}(\Psi^{m-1}+\Psi^{m+1}) = \frac{\sigma_s}{2}\int\limits_{-1}^{1}\gamma^m\Psi^m d\mu', \\ \Psi^m|_{z=0}=0,\ \mu>0; \quad \Psi^0|_{z=H}=1, \Psi^m|_{z=H}=0,\ m\ge1,\ \mu<0; \end{cases} \tag{6}$$

$\Psi^m(z,\mu,p_x)$ is the complex function, i.e. $\Psi^m = \Psi^m_R + i\Psi^m_I$. By analyzing (6) we can to show that $\Psi_R^{2k+1}=0$ и $\Psi_I^{2k}=0$, $k=0,1,\dots,[M/2]$.

By eliminating those components from (6) and defining the new real function $\Phi^m(z,\mu,p_x)$ according to the rule $\Phi^{2k}=\Psi_R^{2k}$, $\Phi^{2k+1}=\Psi_I^{2k+1}$ we find

$$\begin{cases} \mu\frac{\partial\Phi^m}{\partial z}+\sigma\Phi^m+(-1)^m p_x\sqrt{1-\mu^2}(\Phi^{m-1}+\Phi^{m+1}) = \frac{\sigma_s}{2}\int\limits_{-1}^{1}\gamma^m\Phi^m d\mu', \\ \Phi^m|_{z=0}=0,\ \mu>0;\ \Phi^0|_{z=H}=1, \Phi^m|_{z=H}=0,\ m\ge1,\ \mu<0; \end{cases} \tag{7}$$

Expanding $\Phi^m(z,\mu,p_x)$ in Legendre polynomials

$$\Phi^m = \sum_{k=m}^{Nm}\frac{2k+1}{2}\Phi_k^m(z,p_x)Y_k^m(\mu), \tag{8}$$

Substituting this expansion in (7) we obtain the system of ordinary differential equations

$$A\frac{\partial\vec{\Phi}}{\partial z} + [C(z)+p_x D]\vec{\Phi} = 0, \tag{9}$$

A, C(z) are the block diagonal matrices of the order $N = \sum\limits_{m=0}^{M}(N_m-m+1)$. Every block $A_{m,m}$ is the symmetric and triadiagonal matrix of the order N_m-m+1 with zero main diagonal. $C_{m,m} = \mathrm{diag}\{\sigma(2m+1)-\sigma_s g_m, \sigma(2m+3)-\sigma_s g_{m+1},\dots, \sigma(2N_m+1)-\sigma_s g_{N_m}\}$. D is the block-triadiagonal matrix with zero blocks in the main diagonal and blocks $D_{m,m+1}=-D^T_{m+1,m}$ are the rectangular matrices $(N_m-m+1)\times(N_{m+1}-m)$. $\vec{\Phi}=(\vec{\Phi}^0,\vec{\Phi}^1,\dots,\vec{\Phi}^M)^T$. $\vec{\Phi}^m = (\Phi^m_m,\Phi^m_{m+1},\dots,\Phi^m_{N_m})^T$.

For approximation of the boundary conditions in (7) we use Marshak's conditions

$$G_1^m\vec{\Phi}^m(0)=0; \quad G_2^0\vec{\Phi}^0(H)=\vec{1},\ G_2^m\vec{\Phi}^m(H)=0,\ m\ge1, \tag{10}$$

where G_1^m, G_2^m are defined in [2].

For inhomogeneous in the vertical direction atmospheric models, the usual approximation is to divide the atmosphere into several homogeneous layers. In each layer $[z_{l-1},z_l]$ matrix C(z) is the constant and equal C_l. Integrating (9) over z in each layer, we get

$$-\vec{\Phi}(z_{l-1}) + \exp(B\,\Delta z_l)\,\vec{\Phi}(z_l) = 0, \tag{11}$$

where $B=A^{-1}R$, $R=(C_i+p_xD)$.

In order to define matrix $\exp(B\Delta z_i)$ it is necessary to solve eigenvalue problem for matrix B, i.e.

$$R\vec{\beta} = \lambda A\vec{\beta}. \qquad (12)$$

Matrix B has N different complex eigenvalues, which occur in ± pairs,i.e. $\pm\lambda_j$, $j=1,\dots,N/2$, and therefore the order of the problem (12) can be reduced to N/2. For this purpose we define the unitary transformation P_o, which sorts a vector into its odd and even parts, i.e.

$$P_o\vec{\beta} = (\beta_1,\beta_3,\dots,\beta_{N/2-1},\beta_2,\beta_4,\dots,\beta_{N/2})^T = \\ = \vec{\eta} = (\vec{\eta}_t,\vec{\eta}_b)^T .$$

With this substitution, our problem can be written as

$$P_oRP_o^T\vec{\eta}=\lambda P_oAP_o^T\vec{\eta} \quad \text{or} \quad \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}\begin{pmatrix} \vec{\eta}_t \\ \vec{\eta}_b \end{pmatrix}=\lambda\begin{pmatrix} 0 & A_1 \\ A_2 & 0 \end{pmatrix}\begin{pmatrix} \vec{\eta}_t \\ \vec{\eta}_b \end{pmatrix}.$$

Eliminating $\vec{\eta}_t$, we find

$$Z\vec{\eta}_b = \lambda^2\vec{\eta}_b, \quad \text{where } Z=A_1^{-1}R_1A_2^{-1}R_2. \qquad (13)$$

The problem (13) is successfully computing by using the program DHQR2 from the EISPACK collection [3]. Thus we have $B=U\Lambda U^{-1}$, where Λ is the block-diagonal matrix with the blocks $\begin{pmatrix} \lambda_R & \lambda_I \\ -\lambda_I & \lambda_R \end{pmatrix}$, U is the matrix of the eigenvectors. Then $\exp(B)=U\exp(\Lambda)U^{-1}$, where $\exp(\Lambda)$ is the block-diagonal matrix with the blocks $\exp\lambda_R\begin{pmatrix} \cos\lambda_I & \sin\lambda_I \\ -\sin\lambda_I & \cos\lambda_I \end{pmatrix}$.

Thus, the coefficients of (11) are defined. Adding the boundary conditions (10), we obtain the linear system of algebraic equations.

Substituting the computed moments to (8) and then to (4), we obtain some function $\Psi_N(z,\mu,\varphi)$, which oscillates over angle variables around exact solution. When M and N_m are increasing, Ψ_N is slowly converging to Ψ. Therefore, for obtaining of reasonable solution without largely increasing of the order of approximation there is necessary to make smoothing of the spherical harmonics solution.

For this purpose we use the smoothing procedure, obtained by us [2] for the problem (2). The main idea of this method is the numerical evaluation of the error $W = \Psi_N-\Psi$. Following this method we can construct boundary problem for $W(z,\mu,\varphi)$

$$\begin{cases} \mu\dfrac{\partial W}{\partial z} + [\sigma-ip_x\sqrt{1-\mu^2}\cos\varphi]W = S\,W + Q(z,\mu,\varphi), \\ W|_{z=0}=\Psi_N(0,\mu,\varphi),\mu>0; \quad W|_{z=H}=\Psi_N(H,\mu,\varphi)-1,\mu<0; \end{cases} \qquad (14)$$

where $Q(z,\mu,\varphi) =$

$$\sum_{m=0}^{M}(1+\delta_{0m})^{-1}\sqrt{(N_m-m+1)(N_m+m+1)}\frac{d\Phi^m_{N_m}}{dz}\,Y^m_{N_m+1}(\mu)\cos m\varphi + \\ + \frac{ip_x}{2}\sum_{m=1}^{M-1}\Big[\sqrt{(N_m-m)(N_m-m+1)}\;\Phi^{m+1}_{N_m}\;Y^m_{N_m+1}(\mu) + \\ + \sqrt{(N_m-m+1)(N_m-m+2)}\;\Phi^{m+1}_{N_m+1}\;Y^m_{N_m+2}(\mu)\Big]\cos m\varphi \quad -$$

truncating error of the system (9).

Integrating (14) without SW in right hand side we obtain $W_1(z,\mu,\varphi)$, which is the approximation of W in the "single scattering". In order to evaluate W_1 when $\mu<0$ it is necessary to compute the integrals

$$\int_0^{\Delta z}\exp(t/\mu)\begin{Bmatrix} \sin bt \\ \cos bt \end{Bmatrix}\Phi^m_{N_m}(t)dt, \quad b=p_x\sqrt{1-\mu^2}\cos\varphi/\mu,$$

which can be obtained by multiplying of the system (9) at $\exp(t/\mu)\begin{Bmatrix} \sin bt \\ \cos bt \end{Bmatrix}$, by integrating over z and performing respectively matrix transformations. Function W_1 is the exact value of the error of the spherical harmonics approximation of the function Ψ_1, which is the component of the solution of the problem (3) and described "single reflection" from the spreading surface, i.e. Ψ has following form: $\Psi = \Psi_1 + \chi$, where

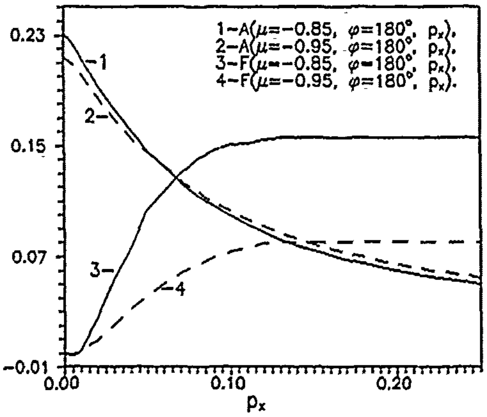
$$\Psi_1=\exp\Big[\int_z^H\sigma(\xi)d\xi/\mu\Big]\big[\cos b(H-z)-i\sin b(H-z)\big]. \qquad (15)$$

$\chi(z,\mu,\varphi)$ characterizes diffuse field and is the smooth function. As seems from (15) function Ψ_1 is the oscillating when $p_x\neq 0$ and therefore is bad approximated by spherical harmonics. Accounting of error W_1 as $\tilde{\Psi} = \Psi_N - W_1$ could described Ψ_1 with high accuracy .

Thus, the describing smoothing procedure will make possible to use the spherical harmonics for solution the problem (3) with high efficiency.

Results of the solving of the problem (3) for the Elterman's atmospheric model [4] for λ=0.75 mkm are presented in figure. Here plotted function χ at the top of atmosphere, which usually presented in following form:

$$\chi(0,\mu,\varphi,p_x)=\exp(iHb)A(\mu,\varphi,p_x)\exp[iF(\mu,\varphi,p_x)].$$



REFERENCES

1. I.V.Mishin,T.A.Sushkevich. Issledovanie Zemli iz cosmosa N6,69,1980.
2. T.Z.Muldashev,U.M.Sultangazin.Gurnal vychislitelnoj matematiki i matematicheskoj fiziki v.26,882,1986.
3. B.T.Smith,J.M.Boyle,J.J.Dongarra,B.S.Garbow I.Ikebe,V.C.Klema and C.B.Moler. Matrix Eigen-system Routines-EISPACK Guide.Springer -Verlag,Berlin (1976).
4. L.Elterman.UV.Visible and IR Attenuation for Altitudes to 50 km. Environmental Research Paper 285.AFCRL.Bedford,MA.1968.

Numerical Solving of Boundary-Value Problems of Mathematical Physic with Reproduction of Solution Group

A.S.Shvedov
Keldysh Institute of Applied Mathematics,
the USSR Academy of Sciences,
Miusskaja sq.4,Moscow,125047,USSR)

The synthesis of simplest algorithms for numerical solution of boundary-value problems with two or three space variables on the base of algorithms for corresponding single-dimensional problems is generally connected only with overcoming technical difficulties.But the quality of such simplest algotithms often proves to be low.For instance, the solution symmetry disturbunce or the approximation loss can occur when the vector field is turned.The synthesys of more perfect algorithms for multidimensional problems is connected with overcoming serious mathematical difficulties.

Let's consider the system of equations for unsteady inviscous compressible fluid flows with three spatial variables.

$$\frac{\partial \rho}{\partial t} + div\ \rho\vec{u} = 0, \quad \frac{\partial e}{\partial t} + div\ e\vec{u} + div\ p\vec{u} = 0, \tag{1}$$
$$\rho\frac{\partial \vec{u}}{\partial t} + grad\ p + \rho\ grad\frac{(\vec{u},\vec{u})}{2} + \rho\ rot\ \vec{u}\times\vec{u} = 0.$$

ρ- density,p- pressure, $\vec{u}$ - velocity of gas, $e = \rho(\varepsilon + 0,5\ |\vec{u}|^2)$, ε - internal energy.Equation of state for p,ρ and ε must be added to the system (1).

THEOREM.*Difference scheme of Godunov's type with strong symmetry conservation property is constructed for system (1).*

The algorithm preserves the symmetry if it reproduces the property of the initial boundary-value problem solution to transform in the solution of the same problem under the effect of shift or rotation groups. At the same time the same algorithm must be adaptable to any group irrespective of whether the group is the shift or rotation group and also irrespective of the shift direction and location of rotation centres or axes. I.e. whatever shift or rotation group of the initial value-boundary problem withstands the numerical solution will transform into itself under the effect of the group discrete analogue.

Here we give the strict mathematical definition of the symmetry conservation property,distinguishing between the strong and weak symmetry conservation.

The algorithm of boundary-value problem numerical solution consists of the definition of the set,serving as a basis for the approximate solution (for the counting region) and for finding the solution.We shall consider the counting region to be an image of a four-dimensional parallelepiped

$$\Xi = \{\ (x^1, x^2, x^3, x^4):$$
$$0 \le x^1 \le L,\ 0 \le x^2 \le M,\ 0 \le x^3 \le N,\ 0 \le x^4 \le K\}$$

(L,M,N,K are positive integers).The piecewise-smooth one-to-one intra Ξ mapping $Y \to \mathbb{R}^4$ has the following form:

$x=x(x^1,x^2,x^3,x^4), y=y(x^1,x^2,x^3,x^4),$
$z=z(x^1,x^2,x^3,x^4), t=t(x^4).$

Here x,y,z are the cartesian coordinates in the space $\mathbb{R}^3$,t is the time.We shall write t_k instead of $t(k)$,$t_0 < t_1 < \ldots < t_K$.

We consider the mapping Y to be linear over x^4 for each line segment

$$k - 1 < x^4 < k, \quad k = 1, \ldots, K.$$

We use Ξ_{lmnk} to designate the three-dimensional cube

$$\{\ (x^1, x^2, x^3, x^4):$$
$$l - 1 \le x^1 \le l,\ m - 1 \le x^2 \le m,$$
$$n - 1 \le x^3 \le n,\ x^4 = k\}.$$

The approximate solution is a function defined on the set $Y(\Xi)$ and constant on every set $Y(\Xi_{lmnk})$, $1 \le l \le L$, $1 \le m \le M$, $1 \le n \le N$, $0 \le k \le K$ .We define the value of the approximate solution on this set as f_{lmnk}.

Definition 1. The boundary-value problem is called two-dimensional (one-dimensional) in the following case. First, there is such a α, β, γ system of rectangular cartesian, or cylindrical , or spherical coordinates at the space $\mathbb{R}^3$ that the problem solution can be defined as a function of the variables α and β (the variable α). Second, if vector fields are used in analysis, the decomposition of these fields into unit vectors of the coordinate system α, β, γ the components that correspond to the unit vector of the variable γ (or unit vectors of the variables β and γ) are to be zero.

Later we shall consider only two- or one-dimensional problems, for which the variables α and β (the variable α) are not angular.It means that the boundary-value problem withstands the effect of the shift or rotation group, and not the extention group.

*Definition 2.*We shall say the grid construction method has the symmetry conservation property, if any two- or one-dimensional problem in the coordinate system α, β, γ appropiate for a given problem,satisfies the equalities

$Y(l,m,n,t_k) = (\alpha_{lm}, \beta_{lm}, \gamma_n, t_k)$ -
- for two-dimensional problems and
$Y(l,m,n,t_k) = (\alpha_l, \beta_m, \gamma_n, t_k)$ -
- for one-dimensional problems.Here

$$\alpha_0 < \alpha_1 < \ldots < \alpha_L,\ \beta_0 < \beta_1 < \ldots$$
$$\ldots < \beta_M,\ \gamma_0 < \gamma_1 < \ldots < \gamma_N.$$

*Definition 3.*The numerical solution algorithm for a class of boundary-value problems with the three space variables has the weak symmetry conservation property, if the grid construction method has the symmetry conservation property and the numerical solution of any two-dimensional problem of the class considered, based on this algorithm satisfies the condition

$$f_{l\,m\,1\,k} = \dots = f_{l\,m\,N\,k} = f_{l\,m\,k};$$

$$1 \le l \le L,\ 1 \le m \le M,\ 0 \le k \le K;$$

and the numerical solution of any one-dimensional problem of the class considered satisfies the condition

$$f_{l\,1\,1\,k} = \dots = f_{l\,M\,1\,k} =$$
$$= f_{l\,1\,2\,k} = \dots = f_{l\,M\,2\,k} = \dots = f_{l\,M\,N\,k} =$$
$$= f_{l\,k};\ 1 \le l \le L,\ 0 \le k \le K.$$

If $f_{l\,m\,n\,k}$ contains vectors, then we consider the vector equality as an equality of their components obtained by their decomposition into unit vectors of the coordinate system α, β, γ.

Definition 4. The numerical solution algorithm for a class of boundary-value problems with the three space variables has the strong symmetry conservation property, if it has the weak symmetry conservation property and of any two-dimensional problems of the class considered the numerical solution $f_{l\,m\,k}$ does not depend on the number N and the set $\gamma_0, \gamma_1, \dots, \gamma_N$, and of any single-dimensional problem of the class considered the numerical solution $f_{l\,k}$ doesn't depend on the numbers M and N and the sets $\beta_0, \beta_1, \dots, \beta_M$ and $\gamma_0, \gamma_1, \dots, \gamma_N$.

An illustration will make the difference between weak and strong symmetry conservations clear. Let the decision of boundary-value problem depend on the r coordinate only in the (z,r,ϕ) cylindrical coordinate system. The different grids for this problem solution are presented in Fig.1,2 in the z=*const* plane. These grids have the same cell number in the r direction and the different cell number in ϕ direction. If the algorithm has the weak symmetry conservation property, then the solution will be identical both for the grid shown in Fig.1 and for the grid shown in Fig.2 for all the cell layers satisfying the fixed ϕ. However, the solution $F_1(r)$ obtained using the first grid, can be different from the solution $F_2(r)$ obtained using the second grid. If the algorithm has the strong symmetry conservation property, $F_1(r)$ fits $F_2(r)$.

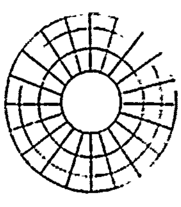


Fig.1

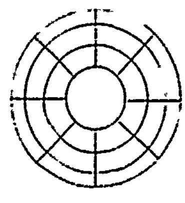


Fig.2

The important property of mathematical physical problems is the independence from the coordinate system, in which they are given since the problem itself does not depend on any coordinate system. In other words, the equations of mathematical physics are written in the invariant vector (or tensor) form. It is the numerical algorithm that is reproduction of this property in analysis. We apply the term "invariance" to the calculation results independence on choice of the x,y,z coordinate system in the space $\mathbb{R}^3$ and parameterization (the mapping Y). Our difference scheme has the invariance property.

The symmetry conservation and invariance properties are important in the numerical solution of complex geometry problems, where the solution can have various kinds of symmetry in various parts of counting region and when it is not clear what kind of coordinate system shall we choose to carry out the analysis. The strong symmetry conservation property is also necessary for solving perturbation problems close to two- or one-dimensional ones. If the numerical algorithm has the strong symmetry conservation property, then we can solve two- and one-dimensional problems using the programs intended for three dimensional calculations.

Three problems are solved for construction of the difference scheme. 1. Construction of curvilinear cells. 2. Obtaining of invariant form of equations of motion close to divergent. 3. Obtaining of formulaes for cell volumes and for cell face areas.

The new method for construction of curvilinear surfaces is proposed. The novelty of the method suggested is that, the countable set of points situated on the surface is generated , to begin with . Then it is proved that, there is the Lipshitz mapping of the parametric plane rectangle into a space $\mathbb{R}^3$ where the binary-rational points of the rectangle are in one-to-one correspondence with the plane points constructed. By this way the surface is parametrically represented.

The multi-dimensional analogue of the Faber - Schauder basis for the space of functions was created for the substantiation of the surface construction.

Three surface families covering the domain so that one and only one surface of each of the families goes through the each point of the domain generate three scalar equations of motion. Each of the equations is generated by one of the surface families. These equations are including only the velocity components obtained by its decomposition into datums that consist of the vectors going normal to or in the main directions of syrfaces. All the geometric entities involved in equations do not depend on ways of specifying or parameterizing surfaces. They are defined by invariants of the first and second quadratic forms. The differentiation is included into the obtained equations only as a divergence of some vector fields.

Proof of the theorem is given in [1].

[1] Shvedov A.S. Difference scheme for gas dynamics equations, conserving group properties of solutions. Matem. zametki, 1990, v.45, N 4, pp.140-151 (Russian). (Transl. in *"Mathematical Notes"*)